

X-643-67-198

NASA TM X-55792

# A METHOD OF CALCULATING INTERPLANETARY TRAJECTORIES

R.C. BLANCHARD  
H.WOLF

MAY 1967

NASA

GODDARD SPACE FLIGHT CENTER  
GREENBELT, MARYLAND

N 67-27579

(ACCESSION NUMBER)

(THRU)

64

(PAGES)

1

TM X-55792

(NASA CR OR TMX OR AD NUMBER)

(CODE)

30

(CATEGORY)

FACILITY FORM 602

1136

X-643-67-198

A METHOD OF CALCULATING INTERPLANETARY TRAJECTORIES

R. C. Blanchard  
and  
H. Wolf

May 1967

Goddard Space Flight Center,  
Greenbelt, Maryland



## CONTENTS

	<u>Page</u>
<b>ABSTRACT . . . . .</b>	v
<b>NOTATION . . . . .</b>	vi
<b>INTRODUCTION . . . . .</b>	1
<b>COASTING ARCS . . . . .</b>	2
<b>TERMINATING CONDITIONS . . . . .</b>	6
Time of Flight . . . . .	6
Periapsis . . . . .	7
Apoapsis . . . . .	8
Distance . . . . .	9
Planet-to-Sun Reference Switch . . . . .	11
Flight-Path-Angle . . . . .	12
Sun-to-Planet Reference Switch . . . . .	13
<b>BURNING ARCS . . . . .</b>	16
<b>ORBITAL MOTION PARAMETERS . . . . .</b>	17
Orbital Elements . . . . .	17
Impact Parameters . . . . .	21
<b>POSITION AND VELOCITY OF PLANETS . . . . .</b>	23
<b>ACKNOWLEDGMENTS . . . . .</b>	26
<b>REFERENCES . . . . .</b>	26
<b>APPENDIX A: F-Functions . . . . .</b>	27
<b>APPENDIX B: Powered Flight Transformation Equations . . . . .</b>	28
<b>APPENDIX C: General Trajectory Supervisor Program (GETSUP) Listings . . . . .</b>	32

## ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	The target planet intercept geometry . . . . .	14
2	The graph of $k$ versus $\beta$ . . . . .	15
3	The impact plane and the relationship of the $\hat{R}$ , $\hat{S}$ , and $\hat{T}$ vectors	22
4	The stretch geometry . . . . .	29
5	The rotation geometry . . . . .	30

A METHOD OF CALCULATING  
INTERPLANETARY TRAJECTORIES

by

R. C. Blanchard  
and  
H. Wolf \*

ABSTRACT

This report presents equations and Fortran program listings for generating approximate patched-conic interplanetary trajectories. The universal form of the conic equations is used for coasting arcs with the universal anomaly,  $\beta$ , as the independent parameter. A pseudo-burn approximation is used for the powered portions of the flight. A selection of terminating conditions for the coasting arcs are presented and discussed. Planet ephemerides are calculated from prestored constants of their motion.

---

\*Analytical Mechanics Associates, Inc., NASA Contract No. NAS5-9298.

## NOTATION

<u>Symbol</u>	<u>Description</u>
a	semi-major axis
A	azimuth
B	impact miss parameter
$(B \cdot \hat{T}), (B \cdot \hat{R})$	impact parameters
$C_0$	$\begin{cases} e \cos E_0, & \text{if } \alpha > 0 \\ e \cosh E_0, & \text{if } \alpha < 0 \end{cases}$
d	$R \cdot \dot{R}$
e	eccentricity
E	$\begin{cases} \text{eccentric anomaly if } \alpha > 0 \\ \text{hyperbola analog to } E \text{ if } \alpha < 0 \end{cases}$
f	true anomaly
$F_i$	i <sup>th</sup> F-function
$G_i$	i <sup>th</sup> G-function
H	angular momentum vector
i	inclination
k	the square of the distance from any given body
m	mass
M	$\begin{cases} \text{mean anomaly if } \alpha > 0 \\ \text{hyperbolic analog to } M \text{ if } \alpha < 0 \end{cases}$
$\hat{P}$	periapsis unit vector
$\hat{Q}$	inplane unit vector perpendicular to $\hat{P}$
r	distance
$r_a$	apoapsis distance
$r_p$	periapsis distance
R, $\dot{R}$	position and velocity vectors, respectively
$\hat{R}, \hat{S}, \hat{T}$	triade describing impact plane
$\hat{S}$	asymptote unit vector

<u>Symbol</u>	<u>Description</u>
$s_0$	$\begin{cases} e \sin E_0, & \text{if } \alpha > 0 \\ e \sinh E_0, & \text{if } \alpha < 0 \end{cases}$
$t$	time
$\Delta t$	time-of-flight
$v$	velocity
$v_s$	circular velocity
$\alpha$	$\equiv 1/a$
$\beta$	universal anomaly, $\equiv \theta / \sqrt{ a }$
$\gamma$	flight-path-angle
$\zeta$	$\equiv -\alpha \beta^2$
$\eta$	mean motion
$\theta$	$\begin{cases} \text{eccentric anomaly difference if } \alpha > 0 \\ \text{hyperbola analog to } E \text{ difference if } \alpha < 0 \end{cases}$
$\omega$	argument of periapsis
$\Omega$	right ascension of ascending node
$\mu$	body's gravitational parameter

### Subscripts

- 0, refers to initial conditions
- 1, refers to updated conditions



## A METHOD OF CALCULATING INTERPLANETARY TRAJECTORIES

### INTRODUCTION

This report presents the equations used in generating approximate interplanetary trajectories. A Fortran program has been written and is called the General Trajectory Supervisor Program (GETSUP). Included in the Appendices is a corresponding program listing. Basically, the GETSUP program generates a patched-conic interplanetary trajectory. The philosophy for calculating the trajectory is that of stringing together in an arbitrary manner a series of arcs, where each arc is thought of as a transformation of the state vector consisting of  $R$ ,  $\dot{R}$ ,  $t$ , and  $m$  with  $\beta$  as the independent variable. The  $\beta$ -parameter is the anomaly-like variable which facilitates a convenient arrangement of the mathematical algorithm for the solution of the universal form of the conic equations.<sup>1, 2, 3, 4, 6, 8</sup> Also, this choice of  $\beta$  as the independent variable has other advantages when calculating interplanetary trajectories. First, many end conditions are geometrical and, therefore, are easily expressed by the  $\beta$ -parameter, or for that matter, any other anomaly-like parameter. Secondly, Kepler's Equation can be solved readily without iterating. These advantages offer some obvious savings in both program design complexities and running time.

There are two types of arcs, coasting arcs and burning arcs. The burning arc transformation is treated as a rotation and/or stretching or contraction of some components of the state vector. The coasting arc transformation of the state, with respect to a given celestial body, is performed by the generalized two-body equations. The desired transformations are specified by the user.

A number of terminating conditions have been incorporated. By proper choice of these conditions, the user is able to obtain the transformed state to some preselected geometrical configuration. The conditions are periapses, apoapsis, a specified distance, flight-path angle, planetary reference switch and  $\beta$ -parameter. Flight-time is also included for completeness. This document describes each end condition and presents the corresponding equations. The programming logic has been designed to incorporate others upon user demand.

The position and velocity of the planets are calculated from prestored constants of the motion which were obtained by utilizing Carpenter's method of generating Chebyshev series coefficients of planetary perturbations.<sup>7</sup> This method included using a differential correction scheme on planetary positions to obtain best fit motion constants. The time range of applicability of the

constants is about twenty years dating from January 10, 1967. This time interval is subdivided into four equal intervals, thus, allowing for a better approximation for the planetary motions.

## COASTING ARCS

The coasting arc is described by a transformation of the state vector with respect to some reference body. Normally, one would transform the state from some time,  $t_0$ , to some other time,  $t_1$ , using time as the independent variable. However, to accomplish this one must solve Kepler's equation by some iterative technique. This problem can be avoided by choosing as the independent variable another parameter which is related to an anomaly. Since, the end conditions are usually defined geometrically, they can be expressed directly as functions of this anomaly rather than as functions of time. Time is considered a component of the state vector since the planetary configuration is given as a function of time. The internal independent parameter used by GETSUP is  $\beta$ , where  $\beta$  is defined as

$$\beta = \frac{E - E_0}{\sqrt{\frac{1}{a}}} = \frac{\theta}{\sqrt{\frac{1}{a}}} \quad (1)$$

Thus given the initial state vector  $(R_0, \dot{R}_0, t_0, m_0)$ , corresponding to  $\beta = 0$  and the body's mass parameter  $\mu$ , equations (2) to (13) are used to obtain the state vector for a given  $\beta$ .

$$r_0 = (R_0 \cdot R_0)^{1/2} \quad (2)$$

$$v_0 = (\dot{R}_0 \cdot \dot{R}_0)^{1/2} \quad (3)$$

$$d_0 = R_0 \cdot \dot{R}_0 \quad (4)$$

$$\frac{1}{a} = \frac{2}{r_0} - \frac{v_0^2}{\mu} \quad (5)$$

Note that the semi-major axis,  $a$ , need not be calculated internally since only its inverse appears in the computations. Also, the sign on  $1/a$  must be preserved in this calculation in order that the equations are applicable to hyperbolic as well as elliptic orbits. The symbol  $\alpha$  is introduced to handle the frequent appearance in the equations of the variable  $1/a$ . Let  $\alpha \equiv 1/a$ .

A parameter is introduced to facilitate the arrangement of the equation into a generalized form. That is

$$\zeta = -\frac{1}{a} \beta^2 = -\alpha \beta^2 \quad (6)$$

As can be seen, for elliptic orbits this parameter in reality is the negative of the squared eccentric anomaly difference, that is,  $\zeta = -(E - E_0)^2 = -\theta^2$ . The F-functions are now defined by

$$F_i = \sum_{n=0}^{\infty} \frac{\zeta^n}{(2n+i)!} \text{ where } i = 0, 1, 2, 3 \quad (7)$$

F-functions are the resulting series expressions for expansions of functions involving trigonometric or hyperbolic functions of the  $\zeta$ -parameter. This arrangement allows one to unify the different conic cases. Appendix A gives a more detailed description. Note, however, that only two of the F-functions need to be calculated by the series, these are  $F_2$  and  $F_3$ . The remaining two are generated from the following recursive expression:

$$F_i = 1 + \zeta F_{i+2} \text{ where } i = 0, 1 \quad (8)$$

Actually, this expression is a particular result of the general recursion formula which is presented in Appendix A. Thus, the G-function utilized in the equations is formed by

$$G_i = \beta^i F_i \text{ where } i = 0, 1, 2, 3$$

For our purposes only the four F-functions are needed, but if partial derivatives are to be computed higher order F-functions are required. The remaining algorithm for position and velocity becomes,

$$r = r_0 G_0 + \frac{d_0 G_1}{\sqrt{\mu}} + G_2$$

$$f = 1 - G_2/r_0$$

$$g = \frac{r_0 G_1}{\sqrt{\mu}} + \frac{d_0 G_2}{\mu} \quad (9)$$

$$\dot{f} = -\frac{\sqrt{\mu} G_1}{r r_0}$$

$$\dot{g} = 1 - G_2/r$$

Thus, the position and velocity vectors of the updated state are

$$\begin{aligned} R &= f R_0 + g \dot{R}_0 \\ \dot{R} &= \dot{f} R_0 + \dot{g} \ddot{R}_0 \end{aligned} \quad (10)$$

The remaining portion of the transformed state, that is  $t$  and  $m$  for  $\beta$  is now considered. The time is calculated by

$$t = t_0 + \Delta t \quad (11)$$

where  $\Delta t$  is the time of flight and is obtained by the universal form of Kepler's equation,

$$\Delta t = \frac{r_0}{\sqrt{\mu}} G_1 + \frac{d_0}{\mu} G_2 + \frac{G_3}{\sqrt{\mu}} \quad (12)$$

Note that during coasting  $m$  remains unchanged. Thus,

$$m = m_0 + \Delta m, \text{ where } \Delta m \equiv 0.$$

For computational purposes the series expressions of  $F_2$  and  $F_3$  are shortened to only 5 terms. The  $\zeta$ -parameter is kept small enough to guarantee the convergence of these series for eight-digit single precision accuracy by utilization of the following  $\zeta$ -parameter reduction formulas

$$F_3(\zeta) = [F_3(\zeta_0) + F_2(\zeta_0) F_1(\zeta_0)]/4 \quad (13)$$

$$F_2(\zeta) = F_1^2(\zeta_0)/2$$

$$\text{where } \zeta_0 = \zeta/4$$

Reduction formulas for  $F_0$  and  $F_1$  also exist but are not needed since they are obtained recursively from  $F_2$  and  $F_3$ . The procedure then is to utilize the reduction equations when  $|\zeta| > 1$ . When  $|\zeta| \leq 1$  and  $\zeta < 0$  then by calculating the first neglected term of each  $F$  series it can be shown that the maximum remainder of  $F_3$  is about  $1.6 \times 10^{-10}$  while for the  $F_2$  series this upper error bound is about  $2.1 \times 10^{-9}$ . When  $\zeta > 0$ , it can be shown that the remainder is essentially unchanged. It is entirely possible that the  $\zeta$ -parameter is very large so that  $\zeta_0$  is not small enough. In this case, the reduction formula is applied repeatedly. After  $n$  such applications,

$$\zeta_0 = \zeta/4n$$

which is less than 1 for sufficiently large  $n$ . Thus  $F_3(\zeta_0)$  and  $F_2(\zeta_0)$  are calculated by the series while the values of  $F_3(\zeta)$  and  $F_2(\zeta)$  are found by utilizing the above reduction equations.

## TERMINATING CONDITIONS

A coasting maneuver is terminated by one of several terminating conditions: a specified time-of-flight, periapsis, apoapsis, given distance, given flight-path-angle, given  $\beta$ , or by switching reference bodies. The updated state vector for any of these configurations can be found quite readily by the equations presented in the previous section providing one has available the proper value of the independent parameter,  $\beta$ . By expressing each of the above end-conditions in terms of  $\beta$  one is able to transform the initial state ( $R_0$ ,  $\dot{R}_0$ ,  $t_0$ ,  $m_0$ ) to the desired final state configuration. Clearly, if the state configuration were desired at some  $\beta$ , then one would be able to use the equations (2) to (12) directly. The following sections give the equations which are used by GETSUP to find the corresponding  $\beta$  for the remainder of the above mentioned conditions. The equations of these conditions do not hold when  $\alpha$  is identically zero, that is, for the parabolic conic. However, it is highly unlikely that this situation would occur. For near parabolic orbits, the resulting  $\beta$  obtained presents no particular problems since the coasting arc calculations are done by the universal form of the two-body equations. It is of course quite simple to express all the conditions mentioned in terms of  $\beta$  by the use of Equations 9, 10, 12 and solve the resulting transcendental equation iteratively and so get perfect generality. However, by specializing to either elliptic or hyperbolic cases we can take advantage of well known expansions of the inverse functions (such as arctangent and logarithm functions) and thus shorten the computation time materially. In view of a great number of trajectories, the resulting program complication seems a reasonable price to pay. In any case, if the parabolic case ever should prove troublesome, equations are available.

### Time-of-Flight

The corresponding  $\beta$  for a specified time of flight,  $\Delta t_0$ , is found by an iteration scheme on the universal form of Kepler's equation. The scheme used by GETSUP is basically a Newtonian iteration suggested by Battin.<sup>6</sup> Nothing is gained here by distinguishing between types of conics except in order to obtain a starting guess which assures convergence of the iteration.<sup>5</sup> This may be done in the following manner: From Equation (5)  $\alpha$  is calculated. Equations (18), given in the next section can be used to calculate  $E_0$ . If  $\alpha$  is negative, then the conic is hyperbolic and the initial guess for  $\beta \equiv \beta_0$ , is

$$\beta_0 = \frac{-E_0}{\sqrt{|\alpha|}} \quad (14)$$

If  $\alpha > 0$ , the circular velocity at  $r_0$  is calculated as

$$v_s = \sqrt{\frac{\mu}{r_0}}$$

Then if the initial velocity magnitude  $v_0$  is in the neighborhood of  $v_s$  i.e.,  $|1 - v_s/v_0| < \epsilon$  where  $\epsilon \ll 1$ , the initial value of the  $\beta$  parameter becomes

$$\beta_0 = \alpha \sqrt{\mu} \Delta t_0 \quad (15)$$

When  $\Delta t_0$  is the desired time-of-flight. For elliptical conics,

$$\beta_0 = \frac{(\pi - E_0)}{\sqrt{\alpha}} \quad (16)$$

With the initial value of  $\beta_0$ , the following sequence of equations is used to find an improved value of  $\beta$ , namely  $\beta_1$ :

$$\begin{aligned} \sqrt{\mu} \Delta t &= r_0 G_1(\beta_0) + \frac{d_0}{\sqrt{\mu}} G_2(\beta_0) + G_3(\beta_0) \\ \frac{\partial(\sqrt{\mu} \Delta t)}{\partial \beta} &= \frac{d_0}{\sqrt{\mu}} G_1(\beta_0) + (1 - \alpha r_0) G_2(\beta_0) + r_0 \\ \beta_1 &= \beta_0 - \left[ \frac{(\sqrt{\mu} \Delta t) - (\sqrt{\mu} \Delta t_0)}{\frac{\partial}{\partial \beta} (\sqrt{\mu} \Delta t)} \right] \end{aligned} \quad (17)$$

The value of  $\beta$  can be improved by passing through Equations (17) until no appreciable change is obtained from one  $\beta$  to the next.

### Periapsis

The determination of the value  $\beta$  to reach periapsis is done somewhat more efficiently by distinguishing among the types of conics. The sign on  $\alpha$  is used for

this purpose. The equations are as follows:

$$S_0^2 = \frac{d_0^2 \alpha}{\mu}$$

$$S_0 = \frac{d_0}{\sqrt{\mu}} |\alpha|^{1/2}$$

$$C_0 = 1 - \alpha r_0$$

$$e = (S_0^2 + C_0^2)^{1/2} \quad (18)$$

$$E_0 = \begin{cases} \tan^{-1} S_0/C_0, & \text{if } \alpha > 0 \\ \ln \left( \frac{S_0 + C_0}{e} \right), & \text{if } \alpha < 0 \end{cases}$$

$$\beta = \frac{-E_0}{|\alpha|^{1/2}}$$

Note that if both  $\alpha$  is negative and  $E_0$  is positive periapsis cannot be reached in positive time. Also  $S_0^2$  is calculated in order that the eccentricity,  $e$ , can be expressed in one equation for both hyperbolic as well as elliptic orbits.

### Apoapsis

In determining the corresponding value of  $\beta$  to reach apoapsis, it is convenient to check the sign of  $\alpha$  immediately since negative values exclude apoapsis possibilities. If  $\alpha$  is positive,  $\beta$  is calculated as follows:

$$\begin{aligned}
 S_0 &= \frac{d_0}{\sqrt{\mu}} \alpha^{-1/2} \\
 C_0 &= 1 - \alpha r_0 \\
 E_0 &= \tan^{-1} \frac{S_0}{C_0}, \quad -\pi \leq E_0 \leq \pi \\
 \beta &= \frac{\pi - E_0}{\alpha^{1/2}}
 \end{aligned} \tag{19}$$

### Distance

Obtaining the corresponding  $\beta$  for terminating a coasting arc at a specified distance,  $r$ , is facilitated by distinguishing among the types of conics. It is clear that if the orbit were circular, then the distance termination would become meaningless. GETSUP provides the user with some diagnostic checks. These checks are particularly helpful for this terminating condition, since there exists a built-in ambiguity. That is, there are normally two distances which satisfy the conic equations. The resolution of these distances is provided for in GETSUP. Equations (20) to (27) handle the distance termination  $\beta$  selection.

$$\begin{aligned}
 S_0^2 &= \frac{d_0^2}{\mu} \alpha \\
 S_0 &= \frac{d_0}{\sqrt{\mu}} |\alpha|^{1/2} \\
 C_0 &= 1 - \alpha r_0 \\
 e &= (S_0^2 + C_0^2)^{1/2} \\
 E_0 &= \begin{cases} \tan^{-1} \frac{S_0}{C_0}, & \text{if } \alpha > 0 \\ \ln \left( \frac{S_0 + C_0}{e} \right), & \text{if } \alpha < 0 \end{cases} \\
 C &= \frac{1}{e} (1 - \alpha r)
 \end{aligned} \tag{20}$$

Note that  $r$  cannot be physically reached under the following conditions

$$\begin{aligned} & \text{if } \alpha > 0 \text{ and } |C| > 1 \\ & \text{if } \alpha < 0 \text{ and } C < 1 \end{aligned} \quad (21)$$

This can readily be seen if one realizes that for the elliptic case,  $C = \cos(E)$  and for the hyperbolic,  $C = \cosh(E)$ . Assuming that the above restrictions are not violated and  $r$  can be obtained, then

$$S = |1 - C^2|^{1/2}, \text{ and} \quad (22)$$

$$E' = \begin{cases} \tan^{-1} S/C, & \text{if } \alpha > 0 \\ \ln(C + S), & \text{if } \alpha < 0 \end{cases} \quad (23)$$

Quadrant allocation will always furnish a positive  $E'$ , since  $S$  by Equation (22) is always positive. To obtain the proper value of  $\beta$  for a given  $r$  requires an adjustment to the sign, or quadrant, of  $E'$ . This criterion for selecting the proper sign is obtained from the following sign function:

$$\sigma_r = \operatorname{sgn}(r - r_0) \quad (24)$$

Notice that when going from an inner to an outer position  $\sigma_r$  is positive while transferring from an outer to an inner position  $\sigma_r$  is negative. This results in the following logic for elliptic transfers. When  $\sigma_r$  is positive:

$$E' = \begin{cases} \geq 0, & \text{if } E_0 \geq 0 \\ E' + 2\pi, & \text{if } E_0 < 0 \end{cases} \quad (25)$$

When  $\sigma_r$  is negative:

$$E' = \begin{cases} \leq 0, & \text{if } E_0 \geq 0 \\ < 0, & \text{if } E_0 < 0 \end{cases} \quad (26)$$

For hyperbolic cases the sign of  $E'$  is that of  $E_0$ . However, if  $\sigma_r < 0$  and  $E_0 > 0$  then the position cannot be physically reached. With the sign of  $E'$  resolved,  $\beta$  is obtained by the following:

$$\beta = \frac{E' - E_0}{|\alpha|^{1/2}} \quad (27)$$

#### Planet-to-Sun Reference Switch

The identical equations of the distance terminator are used to find the corresponding  $\beta$  to terminate the trajectory at the sphere-of-influence of a given planet. This is done by merely letting the distance be equal to the radius of the sphere-of-influence of that particular planet. With this  $\beta$ , the state vector ( $R$ ,  $\dot{R}$ ,  $t$ ,  $m$ ) at the sphere is found by the method presented in Section II. A point transformation can then be made to obtain the state vector with respect to the sun. This involves the following:

$$\begin{aligned} R_s &= R + R_N \\ \dot{R}_s &= \dot{R} + \dot{R}_N \\ m_s &= m \\ t_s &= t \end{aligned} \quad (28)$$

Where  $R_N$  and  $\dot{R}_N$  are the position and velocity vectors of the planet at the time  $t$ . A discussion of the method of obtaining the planet's position and velocity is presented later in the text.

### Flight-Path-Angle

The equations used to determine the corresponding value of  $\beta$  to reach a specified flight-path-angle,  $\gamma$ , are presented in this section. An ambiguity appears for elliptic orbits since each flight-path-angle occurs twice. The equations presented give the  $\beta$  for flight-path-angles near periapsis. The equations are as follows:

$$\begin{aligned} S_0^2 &= \frac{d_0^2}{\mu} \alpha \\ S_0 &= \frac{d_0}{\sqrt{\mu}} |\alpha|^{1/2} \end{aligned} \quad (29)$$

$$C_0 = 1 - \alpha r_0$$

$$e = (C_0^2 + S_0^2)^{1/2}$$

$$E_0 = \begin{cases} \tan^{-1} \frac{S_0}{C_0}, & \text{if } \alpha > 0 \\ \ln \frac{S_0 + C_0}{e}, & \text{if } \alpha < 0 \end{cases} \quad (30)$$

$$S = \frac{\sqrt{|1 - e^2|}}{e} \tan \gamma \quad (31)$$

$$C = \begin{cases} \sqrt{1 - S^2}, & \text{if } \alpha > 0 \\ \sqrt{1 + S^2}, & \text{if } \alpha < 0 \end{cases} \quad (32)$$

$$E = \begin{cases} \tan^{-1} S/C, & \text{if } \alpha > 0 \\ \ln (S + C), & \text{if } \alpha < 0 \end{cases} \quad (33)$$

$$\beta = \frac{E - E_0}{|\alpha|^{1/2}} \quad (34)$$

Notice that when  $|S| > 1$  and  $\alpha > 0$ , the flight-path-angle cannot be reached. Also, when  $\alpha < 0$  and  $E_0 > E$ , the flight-path-angle cannot be reached.

#### Sun-to-Planet Reference Switch

The method and equations used in obtaining the proper value of  $\beta$  in sun reference for terminating at the sphere-of-influence of a given planet are presented in this section. In general, the problem of finding this  $\beta$  involves an iteration scheme, since the geometrical coupling of the probe's position in the orbit with the planet's ephemerides makes it difficult to give an explicit expression for  $\beta$ . A direct scheme might consist of selecting an incremental beta,  $\Delta\beta$ , and updating the state step-wise, saving the values of  $\beta$  before and after each update. At each  $\beta$ -step the quantity  $k$  can be computed by

$$k = R' \cdot R', \text{ where } R' = R - R_N \quad (35)$$

$R$  is the position vector of the satellite with respect to the sun and is obtained by the equations (2) through (13).  $R_N$  is position of the target planet with respect to the sun.  $R'$  is the position vector of the satellite with respect to the new reference body, namely the target planet. This vector is obtained by the equations (66) through (68).  $k$  can then be compared with  $k_s$ , where

$$k_s = r_s^2, \text{ and} \quad (36)$$

$r_s$  = radius of sphere-of-influence of the target planet

Once  $k < k_s$ , that is the proper value of  $\beta$  has been bracketed, an improved value can be found by,

$$\beta_N = \beta_{N-2} - (k_{N-2} - k_s) \left[ \frac{\beta_{N-1} - \beta_{N-2}}{k_{N-1} - k_{N-2}} \right] \quad (37)$$

A basic problem with this scheme is selecting the correct step size,  $\Delta\beta$ . When  $\Delta\beta$  is too large, the linear iterator will not be very effective. A small value of  $\Delta\beta$  may result in many steps before bracketing the solution. It is also bothersome

to discriminate between other relative minima which may be encountered on the way. Another scheme has been devised to eliminate these difficulties. It involves obtaining a first guess of  $\beta$  near the solution and subsequently evaluating the direction in which to proceed in order to trap the desired value of  $\beta$ . The procedure is to find the corresponding  $\beta$  needed to intersect the planet's mean orbit. This is done by the use of the distance termination, presented earlier, where  $r$  is the mean distance of the target planet from the sun. A schematic of the geometry is given by Figure 1.

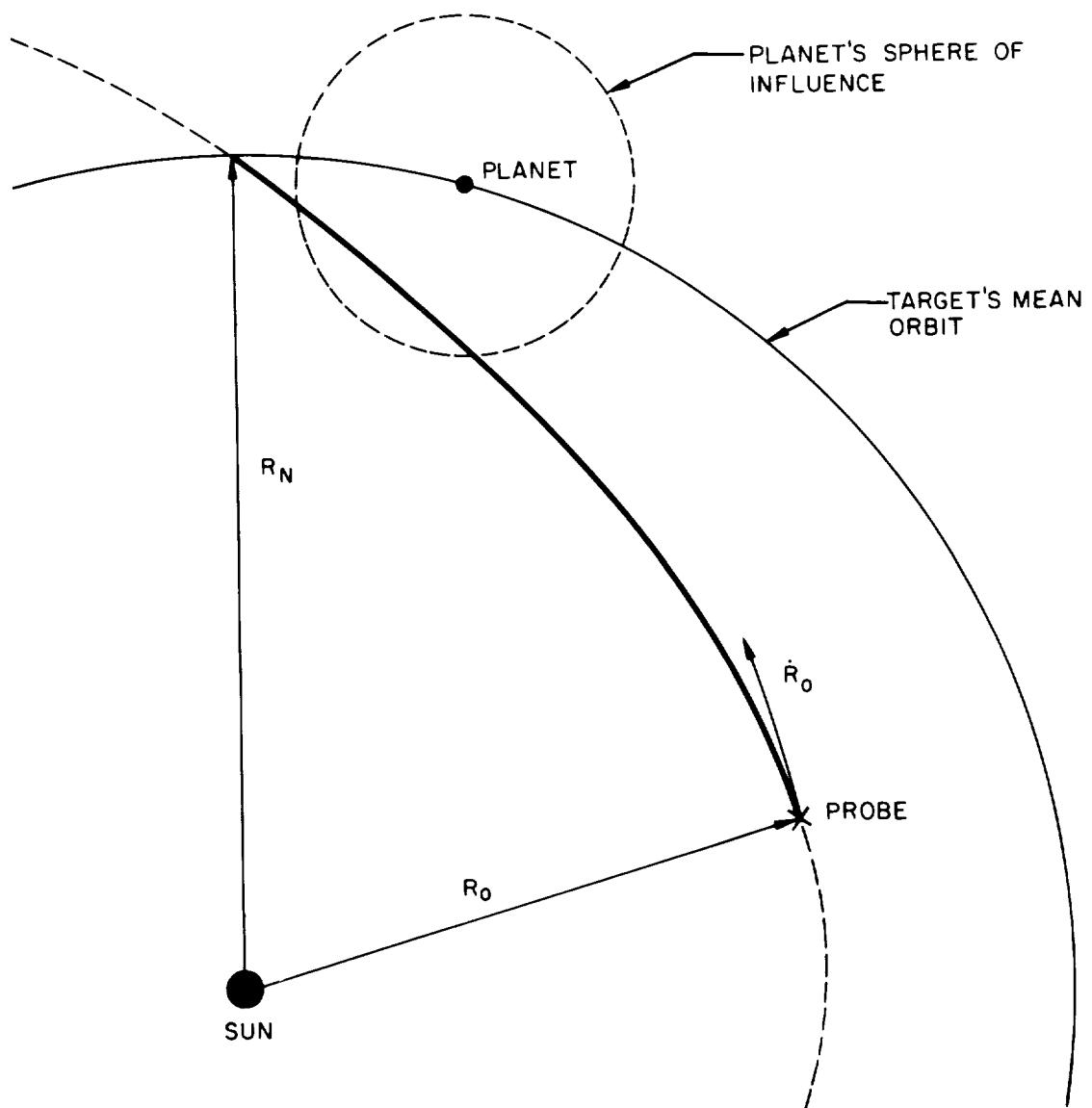


Figure 1—The target planet intercept geometry

The value of  $k$  is computed by Equation (35). In general, the graph of  $k$  versus  $\beta$  is given in Figure 2. Notice that there exist two possible solutions for a given  $k_s$ . This is apparent since the first solution corresponds to entering the planet's sphere while the second corresponds to exiting the sphere. Also, when  $k_s < k_m$ , a solution cannot be found. This means that the trajectory of the probe does not enter the planet's sphere.

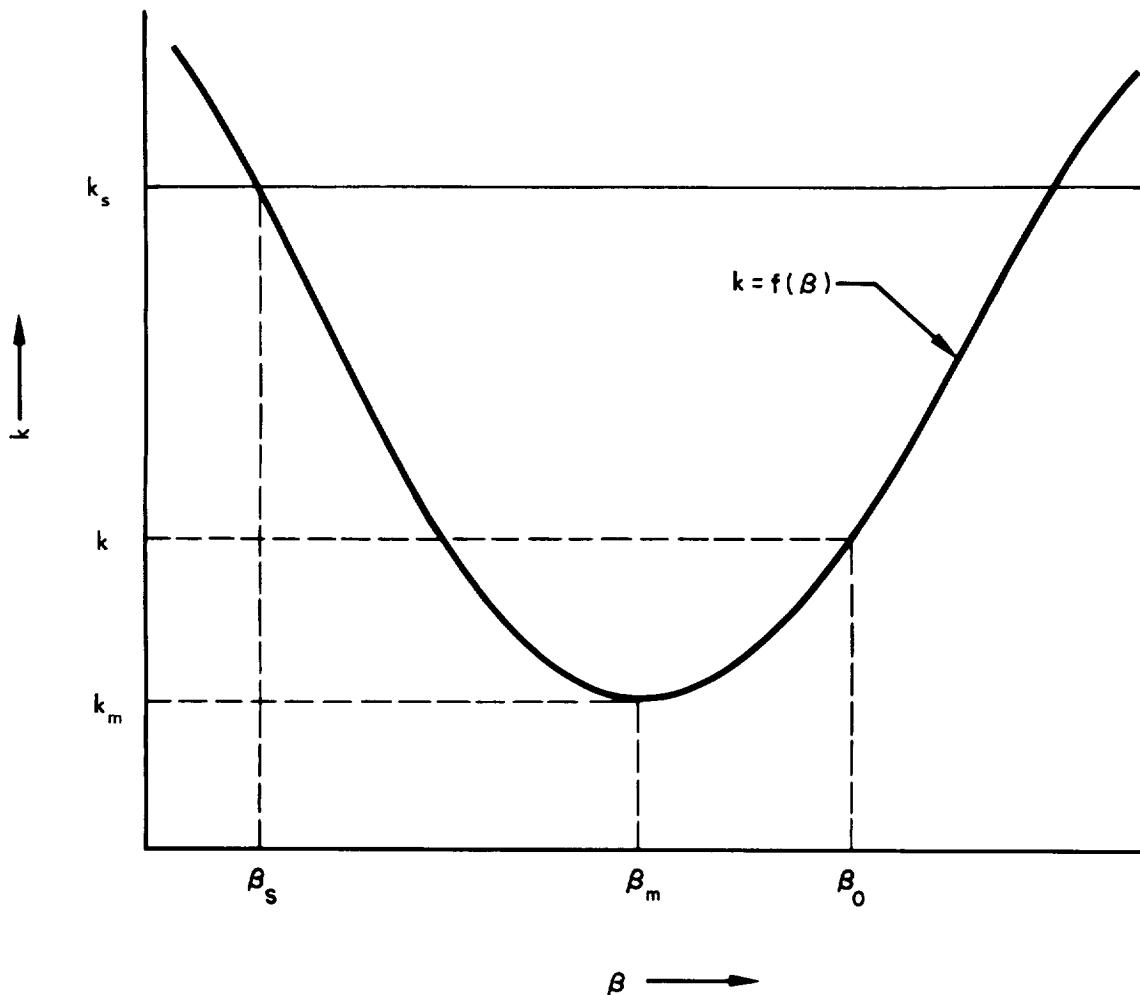


Figure 2-The graph of  $k$  versus  $\beta$

The derivative of  $f(\beta)$  at the point  $\beta_0$  is numerically computed. If the value of the derivative is positive,  $\beta$  is incremented to the left until the value of  $\beta$  which gives  $k_s$  is bracketed. Improved values of  $\beta$  can be found by iterating using Equation (37). If the derivative is negative, then the value of  $k$  at  $\beta_0$  is compared with  $k_s$  and the sign on step-size is selected appropriately in order to bracket the desired solution.

## BURNING ARCS

The powered flight portions of the trajectory are approximated by arcs. These arcs are characterized by corresponding changes in burn-coast angle,  $\Delta s$ , speed,  $\Delta v$ , flight path angle,  $\Delta \gamma$ , flight path azimuth,  $\Delta A$ , height,  $\Delta h$ , time,  $\Delta t$ , and mass,  $\Delta m$ . All of these changes except  $\Delta t$  and  $\Delta m$  are accomplished by rotations and stretchings of the position and velocity components of the state vector,  $(R_0, \dot{R}_0, t_0, m_0)$ . A stretching of a general vector  $Y$  by an amount  $\Delta \ell$  results in the vector  $Y'$  where

$$Y' = \left( \frac{y + \Delta \ell}{y} \right) Y \quad (38)$$

A rotation of  $Y$  about a vector  $A$  through an angle  $\epsilon$  results in a vector  $Y'$  which is given by

$$Y' = \left( 1 - 2 \sin^2 \frac{\epsilon}{2} \right) Y + 2 \left( \frac{A \cdot Y}{a^2} \right) \sin^2 \frac{\epsilon}{2} A + \frac{A \times Y}{a} \sin \epsilon \quad (39)$$

A discussion of these equations is given in Appendix A. The burn rotate and stretch transformation consists now of the following table:

### Stretch:

VECTOR	by AMOUNT
$R_0$	$\Delta h$
$(\dot{R}_0)_{  }$	$\Delta v$

### Rotate:

VECTOR	About VECTOR	ANGLE
$R_0$	$H \equiv R_0 \times \dot{R}_0$	$\Delta s$
$R_0$	$\dot{R}_0$	$-\Delta A$
$\dot{R}_0$	$H$	$\Delta s - \Delta \gamma$

These changes in the position and velocity vectors can be performed separately or in any combination. The remaining portion of the state vector  $m$  and  $t$ , are transformed by

$$t = t_0 + \Delta t \quad (40)$$

$$m = m_0 + \Delta m$$

The quantity  $\Delta m$  can also be computed by

$$x = \frac{\Delta v}{I_s g_0} \quad (41)$$

$$\Delta m = m_0 (1 - e^{-x})$$

where the specific impulse,  $I_s$  and the change of speed  $\Delta v$  are specified by the user.

#### ORBITAL MOTION PARAMETERS

At various times during the coasting arcs it is of interest to express the state vector in other related parameters of the motion. GETSUP calculates two sets of these parameters. They are called the Orbital Elements and the Impact Parameters.

##### Orbital Elements

Given the state vector  $(R_0, \dot{R}_0, t_0, m_0)$  with respect to a given body, specified by the  $\mu$  parameter, equations (42) to (59) are used to obtain the two-body orbital elements.

$$d_0 = R_0 \cdot \dot{R}_0$$

$$r_0 = (R_0 \cdot R_0)^{1/2} \quad (42)$$

$$v_0^2 = \dot{R}_0 \cdot \dot{R}_0$$

$$\alpha = \frac{2}{r_0} - \frac{v_0^2}{\mu}$$

Note here again the sign of  $\alpha$  is preserved to handle hyperbolic as well as elliptic orbits.

$$\text{Semi-major axis; } a = \frac{1}{\alpha} \quad (43)$$

$$\text{Mean motion; } \eta = \sqrt{\mu} |\alpha|^{-1/2} \quad (44)$$

$$C_0 = 1 - \alpha r_0$$

$$S_0 = \frac{d_0}{\sqrt{\mu}} |\alpha|^{1/2} \quad (45)$$

$$S_0^2 = \alpha \frac{d_0^2}{\mu}$$

Eccentricity;

$$e = (C_0^2 + S_0^2)^{1/2} \quad (46)$$

Eccentric anomaly;

$$E_0 = \begin{cases} \tan^{-1} \frac{S_0}{C_0}, & \text{if } \alpha > 0 \\ \ln \left( \frac{C_0 + S_0}{e} \right), & \text{if } \alpha < 0 \end{cases} \quad (47)$$

Note that  $E_0$  is the analogous eccentric anomaly parameter for hyperbolic orbits, that is, when  $\alpha < 0$ .

Mean anomaly;

$$M = \begin{cases} E_0 - S_0, & \text{if } \alpha > 0 \\ S_0 - E_0, & \text{if } \alpha < 0 \end{cases} \quad (48)$$

Time to periapsis;

$$t_p = -\frac{M}{\eta} \quad (49)$$

Angular momentum vector;

$$\mathbf{H} = \mathbf{R}_0 \times \dot{\mathbf{R}}_0 \quad (50)$$

where the components of  $\mathbf{H}$  are  $(h_x, h_y, h_z)$  and the magnitude is obtained by

$$h = (H \cdot H)^{1/2} \quad (51)$$

Periapsis unit vector;

$$\hat{\mathbf{P}} = \frac{1}{e} \left( \frac{1}{r_0} - \alpha \right) \mathbf{R}_0 - \frac{d_0}{e\mu} \dot{\mathbf{R}}_0 \quad (52)$$

where the components of  $\hat{\mathbf{P}}$  are  $(p_x, p_y, p_z)$

$$\cos i = \frac{h_z}{h} \quad (53)$$

$$\sin i = \sqrt{1 - \cos^2 i}$$

**Inclination;**

$$i = \tan^{-1} \left( \frac{\sin i}{\cos i} \right)$$

$$\sin \Omega = \frac{h_x}{h \sin i}$$

$$\cos \Omega = - \frac{h_y}{h \sin i}$$

**Right Ascension of Ascending Node;**

$$\Omega = \tan^{-1} \left( \frac{\sin \Omega}{\cos \Omega} \right) \quad (54)$$

$$\cos \omega = p_x \cos \Omega + p_y \sin \Omega$$

$$\sin \omega = \frac{p_x}{\sin i}$$

**Argument of Periapsis;**

$$\omega = \tan^{-1} \left( \frac{\sin \omega}{\cos \omega} \right) \quad (55)$$

$$C_1 = \sqrt{\frac{1+e}{|1-e|}}$$

**True Anomaly;**

$$f = \begin{cases} 2 \tan^{-1} [C_1 \tan E_0 / 2], & \text{if } \alpha > 0 \\ 2 \tan^{-1} [C_1 \tanh E_0 / 2], & \text{if } \alpha < 0 \end{cases} \quad (56)$$

Periapsis;

$$r_p = a(1 - e) \quad (57)$$

When  $\alpha > 0$ , the following are also calculated

Apofocus;

$$r_a = a(1 + e) \quad (58)$$

Period;

$$\tau = \frac{2\pi}{\eta} \quad (59)$$

### Impact Parameters

The impact, or target, parameters ( $B \cdot \hat{T}$ ) and ( $B \cdot \hat{R}$ ) are usually examined when generating interplanetary targeting trajectories. Consequently, let us assume that the state vector ( $R_0, \dot{R}_0, t_0, m_0$ ) has been switched into the target planet reference and the orbital elements with respect to this planet are available. One is then able to calculate the incoming asymptote unit vector,  $\hat{S}$ , as follows;<sup>8</sup>

$$\begin{aligned} \hat{Q} &= \frac{H}{h} \times \hat{P} \\ \hat{S} &= \frac{1}{e} (\hat{P} + \sqrt{e^2 - 1} \hat{Q}) \end{aligned} \quad (60)$$

where the angular momentum vector,  $H$ , and its magnitude,  $h$ , are calculated by Equation (50) and the periapsis unit vector  $\hat{P}$  is calculated by Equation (52). The impact parameters are coordinates in the impact plane, which is normal to the incoming asymptote. A schematic of the impact plane and the relationship of the  $\hat{R}$ ,  $\hat{S}$  and  $\hat{T}$  vectors, is given in Figure 3.

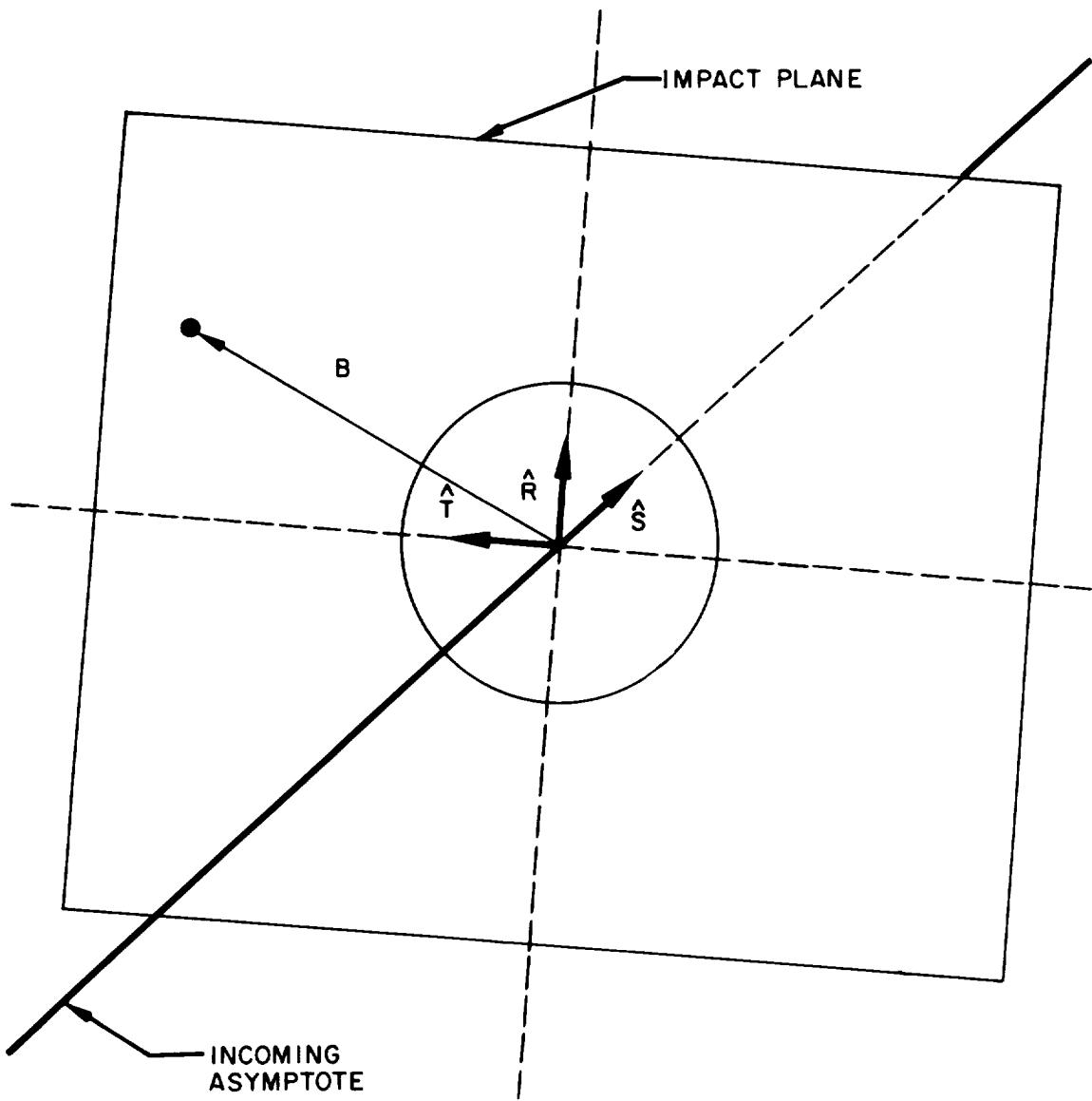


Figure 3—The Impact Plane and the relationship of the  $\hat{R}$ ,  $\hat{S}$ , and  $\hat{T}$  vectors.

$\hat{T}$  is a unit vector tangent to the target planets orbital motion in the ecliptic plane and is obtained as follows:

$$\hat{T} = \frac{\hat{N} \times \hat{S}}{|\hat{N} \times \hat{S}|} \quad (61)$$

where  $\hat{N}$  is the unit normal to the ecliptic plane. The unit vector  $\hat{R}$  is normal to both  $\hat{S}$  and  $\hat{T}$  and is calculated as

$$\hat{R} = \hat{S} \times \hat{T} \quad (62)$$

The magnitude of the miss parameter,  $|B|$  is now calculated as

$$|B| = \frac{a(1 - e^2)}{1 + \sqrt{e^2 - 1}} \quad (63)$$

while

$$\hat{B} = \frac{\sqrt{e^2 - 1}}{e} \hat{P} + \frac{1}{e} \dot{\hat{P}} \quad (64)$$

where  $\hat{P}$  is obtained by Equation (50) and  $\dot{\hat{P}}$  is

$$\dot{\hat{P}} = \frac{d_0}{hr_0e} R_0 + \frac{1}{e} \left( \frac{h}{\mu} - \frac{r_0}{h} \right) \dot{R}_0 \quad (65)$$

The two target components  $(B \cdot \hat{T})$  and  $(B \cdot \hat{R})$  are now calculated where

$$B = |B| \hat{B}$$

## POSITION AND VELOCITY OF PLANETS

The method and equations used in obtaining the position and velocity of the planets are presented in this section. The reference coordinate system is heliocentric cartesian. The method involves using precalculated motion constants obtained from an analysis performed using Carpenter's Chebyshev Series representations of planetary perturbations.<sup>7</sup> The analysis included utilizing the digital programs which represented the planetary perturbations in terms of

Chebyshev series. Examining resulting coefficients of components of the perturbations over a twenty year time period has revealed that it is somewhat efficient to subdivide the interval into four parts. The choice is between the number of coefficients and the desired planetary position accuracy. However, for purposes of quickly generating approximate interplanetary trajectories, it is possible to use only the constants of the motion without the perturbative coefficients providing the time-span is not too large. This eliminates the need of adding the corrective terms due to perturbations, thus eliminating the Chebyshev recursive process. However, when increased accuracy of planetary positions is required, the additional terms due to perturbations by other planets may readily be included. The cartesian coordinates of the planets are accurate to approximately four digits with the present motion constants for the following time intervals;

$$1/10/67 \leq t_1 < 1/8/72$$

$$1/8/72 \leq t_2 < 1/5/77$$

$$1/5/77 \leq t_3 < 1/3/82$$

$$1/3/82 \leq t_4 < 1/1/87$$

where each interval is of length 1824.0 days.

For a given time and planet equations (66) to (68) are used to calculate the position and velocity of that planet from the motion constants  $A_i$ ,  $B_i$ ,  $\eta_i$ ,  $M_{0_i}$ ,  $e_i$  where the subscript i refers to the following celestial body:

- i = 1, SUN
- 3, VENUS
- 4, EARTH
- 6, MARS
- 7, JUPITER
- 8, SATURN
- 9, URANUS
- 10, NEPTUNE
- 11, PLUTO

and where

$$A_i = a \hat{P}_i, \hat{P}_i \text{ is the in-plane vector toward periapsis}$$

$$B_i = b \hat{Q}_i, \hat{Q}_i \text{ is the in-plane vector perpendicular to } \hat{P}_i$$

$$\eta_i = \text{mean motion}$$

$$M_{0i} = \text{mean anomaly at the center of each time period}$$

$$e_i = \text{eccentricity}$$

Notice that the planet Mercury and the Moon are not included in the above list. These are special cases and will be considered separately since the motion of these bodies are quite irregular. The  $\hat{P}$  and  $\hat{Q}$  vectors are calculated using the value for the obliquity of the ecliptic of 23,44579 degrees for Jan. 0, 1950. The mean anomaly at the given time  $t$  becomes,

$$M_i = M_{0i} + \eta_i (t - t_{0i}) \quad (66)$$

where  $t_{0i}$  is the time at the beginning of time interval +912<sup>d</sup>. Kepler's equation,

$$M_i = E_i - e_i \sin E_i \quad (67)$$

is now solved for  $E_i$ . This involves an iterative technique<sup>7</sup> which is convergent since the planet's motion is naturally nearly circular. Thus the position  $R_{p_i}$  and velocity,  $\dot{R}_{p_i}$  of the  $i$ th planet are

$$\begin{pmatrix} R_{p_i} \\ \dot{R}_{p_i} \end{pmatrix} = \begin{pmatrix} \cos E_i - e_i & \sin E_i \\ \frac{\eta_i \sin E_i}{1 - e_i \cos E_i} & \frac{\eta_i \cos E_i}{1 - e_i \cos E_i} \end{pmatrix} \begin{pmatrix} A_i \\ B_i \end{pmatrix} \quad (68)$$

## ACKNOWLEDGMENTS

I am indebted to R. A. Devaney, a former member of the Math and Computing Branch of the Laboratory for Theoretical Studies, for the original development of the Fortran IV digital program.

## REFERENCES

1. Analytical Mechanics Associates, Inc., "Final Report for Minimum Variance Precision Tracking and Orbit Prediction Program," Contract No. NAS5-2535, prepared for Special Projects Branch, GSFC, Greenbelt, Maryland, May 1963.
2. Sperry Corp., "Program Manual for Operational Minimum Variance Tracking and Orbit Prediction Program," prepared for Special Projects Branch, GSFC, Greenbelt, Maryland, January 1964.
3. Analytical Mechanics Associates, Inc., "A Set of Uniform Variational Parameters for Space Trajectory Analysis," Contract No. NAS5-9085, prepared for Special Projects Branch, GSFC, February 1967.
4. Stumpff, K., Himmelsmechanik, VEB Verlag, Berlin, 1959.
5. Lancaster, E. R., private communication.
6. Battin, R. H., Astronautical Guidance, McGraw-Hill, Inc., Copyright 1964.
7. Carpenter, L., "Planetary Perturbations in Chebyshev Series," NASA TN D-3168, January 1966.
8. Shaffer, R., Squires, R. K. and Wolf, H., "ITEM Program Manual," GSFC, X-640-63-71, May 1963.

## APPENDIX A: F-Functions

The F-functions presented in this report are convenient series expressions for expansions of trigonometric or hyperbolic functions of  $\zeta$  where

$$\zeta = -\frac{1}{a} \beta^2 = \mp \theta^2$$

The negative sign is associated with elliptic orbits while the positive sign refers to hyperbolic orbits. References 1, 2, 3, 6 and 8 utilize these series in one form or another to obtain the universal form of the two-body equations. In general,

$$F_i = \sum_{n=0}^{\infty} \frac{\zeta^n}{(2n+i)!} \quad (A-1)$$

In particular,

$$\begin{aligned}
 F_3 &= \frac{1}{3!} + \frac{\zeta}{5!} + \frac{\zeta^2}{7!} + \frac{\zeta^3}{9!} + \dots = \begin{cases} \frac{\theta - \sin \theta}{\theta^3} & , \text{ if } \zeta < 0 \\ \frac{\sinh \theta - \theta}{\theta^3} & , \text{ if } \zeta > 0 \end{cases} \\
 F_2 &= \frac{1}{2!} + \frac{\zeta}{4!} + \frac{\zeta^2}{6!} + \frac{\zeta^3}{8!} + \dots = \begin{cases} \frac{1 - \cos \theta}{\theta^2} & , \text{ if } \zeta < 0 \\ \frac{\cosh \theta - 1}{\theta^2} & , \text{ if } \zeta > 0 \end{cases} \\
 F_1 &= 1 + \frac{\zeta}{3!} + \frac{\zeta^2}{5!} + \frac{\zeta^3}{7!} + \dots = \begin{cases} \frac{\sin \theta}{\theta} & , \text{ if } \zeta < 0 \\ \frac{\sinh \theta}{\theta} & , \text{ if } \zeta > 0 \end{cases} \\
 F_0 &= 1 + \frac{\zeta}{2!} + \frac{\zeta^2}{4!} + \frac{\zeta^3}{6!} + \dots = \begin{cases} \cos \theta & , \text{ if } \zeta < 0 \\ \cosh \theta & , \text{ if } \zeta > 0 \end{cases}
 \end{aligned} \quad (A-2)$$

By writing out the first term, the F-series can also be written as

$$F_i = \frac{1}{i!} + \sum_{n=1}^{\infty} \frac{\zeta^n}{(2n+i)!} \quad (A-3)$$

Letting  $n = k + 1$  and rearranging gives the recursive form for the F-series; that is,

$$F_i = \frac{1}{i!} + \zeta F_{i+2} \quad (A-4)$$

By calculating the highest integer F-series, that is, the largest value of  $i$ , one is then able to compute the remaining lower integer F-series by the above recurrence relationship. This usually halves the amount of series calculations.

It is possible to calculate the F-series functions accurately with only a few terms by limiting the magnitude of  $\zeta$  to less than one. However, this restriction would seriously hinder general conic calculations since the  $\zeta$ -parameter is related to anomaly. A way to avoid this type of difficulty is to test  $\zeta$  and rectify the conic trajectory, that is, update the initial position and velocity vector piecemeal. Another method, and the one used by GETSUP, is to utilize the reduction forms for the F-functions. These are,

$$4F_3(4\zeta) = F_3(\zeta) + F_2(\zeta)F_1(\zeta) \quad (A-5)$$

$$2F_2(4\zeta) = F_1^2(\zeta)$$

The reduction formulas for  $F$  are in reality half-angle formulas; elliptic or hyperbolic depending on the sign of  $\zeta$ . The appearance of the four is due to the fact that  $|\zeta|$  appears in the functions.

## APPENDIX B: Powered Flight Transformation Equations

The burning portion of the trajectory is handled by a stretch and rotation. This appendix develops the corresponding equations. Let an arbitrary vector,  $\mathbf{Y}$ , of length,  $y$ , be stretched by an amount  $\Delta\mathbf{y}$ . Consider the following sketch given in Figure 4.

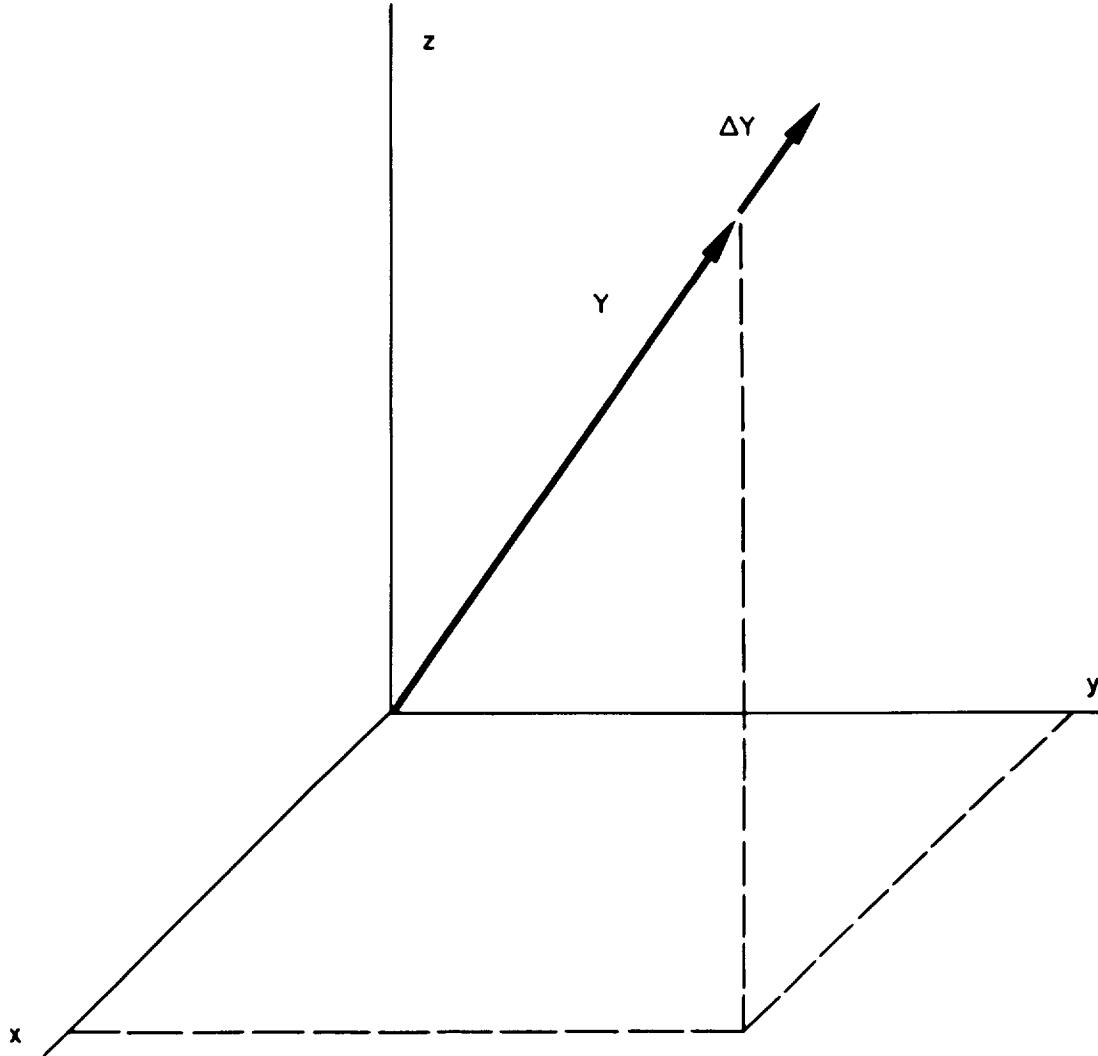


Figure 4—The stretch geometry

It is apparent that the stretched vector  $\mathbf{Y}'$  is

$$\mathbf{Y}' = \mathbf{Y} + \Delta\mathbf{Y} \quad (\text{B-1})$$

where  $\Delta Y$  is the vector parallel to  $Y$  and of length  $\Delta y$ , that is,

$$\Delta Y = \Delta y \frac{Y}{y} \quad (B-2)$$

Or, substituting and rearranging results in

$$Y' = \left( \frac{y + \Delta y}{y} \right) Y \quad (B-3)$$

Notice that the original vector,  $Y$ , can be stretched or shortened by proper use of the sign of  $\Delta y$ .

The rotation of any vector,  $Y$ , about some other arbitrary vector,  $A$ , through an angle,  $\epsilon$ , is now discussed. Consider the following diagram given in Figure 5.

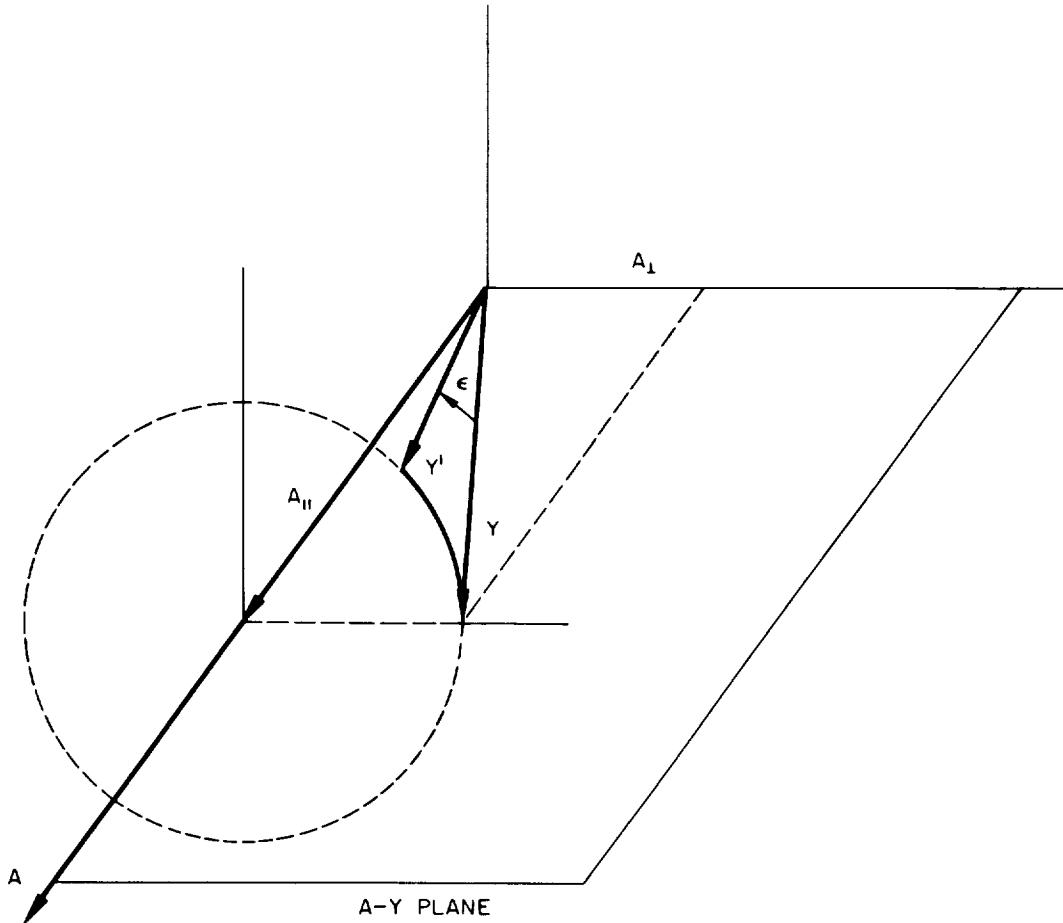


Figure 5-The rotation geometry

The projection of  $\mathbf{Y}$  onto  $\mathbf{A}$  is

$$\mathbf{A}_{||} = \left( \mathbf{Y} \cdot \frac{\mathbf{A}}{a} \right) \frac{\mathbf{A}}{a} \quad (\text{B-4})$$

where  $a$  is the magnitude of  $\mathbf{A}$ . The projection of  $\mathbf{Y}$  onto an axis perpendicular to the  $\mathbf{A}$  axis but in the plane is

$$\mathbf{A}_{\perp} = \mathbf{Y} - \mathbf{A}_{||} = \mathbf{Y} - \left( \mathbf{Y} \cdot \frac{\mathbf{A}}{a} \right) \frac{\mathbf{A}}{a} \quad (\text{B-5})$$

Now  $\mathbf{Y}'$  can be expressed in terms of  $\mathbf{A}_{||}$  and  $\mathbf{A}_{\perp}$  as follows

$$\mathbf{Y}' = \mathbf{A}_{||} + |\mathbf{A}_{\perp}| \cos \epsilon \frac{\mathbf{A}_{\perp}}{|\mathbf{A}_{\perp}|} + |\mathbf{A}_{\perp}| \sin \epsilon \frac{\mathbf{A}}{a} \times \frac{\mathbf{A}_{\perp}}{|\mathbf{A}_{\perp}|} \quad (\text{B-6})$$

Rearranging and substituting gives

$$\mathbf{Y}' = \cos \epsilon \mathbf{Y} + \frac{\mathbf{Y} \cdot \mathbf{A}}{a^2} (1 - \cos \epsilon) \mathbf{A} + \left( \frac{\sin \epsilon}{a} \right) \mathbf{A} \times \mathbf{Y} \quad (\text{B-7})$$

Using the half-angle identities leaves the rotation equations as

$$\mathbf{Y}' = \left( 1 - \sin^2 \frac{\epsilon}{2} \right) \mathbf{Y} + \frac{2(\mathbf{A} \cdot \mathbf{Y})}{a^2} \sin^2 \frac{\epsilon}{2} \mathbf{A} + \frac{\sin \epsilon}{a} \mathbf{A} \times \mathbf{Y} \quad (\text{B-8})$$

## APPENDIX C: General Trajectory Supervisor Program (GETSUP) Listings

The GETSUP program is in a state of constant evolution and, therefore, the mathematical notation and expressions as found in the program listings may vary somewhat from the way in which they are presented in this report. Certain features presented in the theory are still in the development stages and will be appended to the program when they become available. However, the listings and equations are essentially equivalent.

The logic has been designed such that the variables and constants of the program are stored into an array, designated the C-ARRAY. Modification of these constants or variables is provided for by the following input cards:

The first data card contains identification to identify the run. Column 1 of this card must contain a "1."

The next data card contains the control indices used by the program. These are:

COLUMN NOS.	DESCRIPTION
1	* indicates last data card for a particular case while $\Delta^t$ indicates more data to follow.
2-3	arc number, if $\Delta$ then program automatically sequences the arcs.
5	Indicates whether to use position and velocity of last sub-arc; Y indicates "YES," N indicates "NO."
7	Indicates whether new C-array data will follow; Y indicates "YES," N indicates "NO."
9	Flight direction: F for "FORWARD" and B for "BACKWARD."
11	Maneuver Type: C for "COAST" and B for "BURN."

---

<sup>t</sup>  $\Delta$  is the symbol used here to represent a blank column.

COLUMNS NOS.	DESCRIPTION
13	Burn type number where 1 = Rotate R about H through angle $\Delta s$ 2 = Rotate R about H through angle $\Delta s - \Delta \gamma$ 3 = Rotate R about R through angle $-\Delta A$ 4 = Stretch R by $\Delta h$ 5 = Stretch R    by $\Delta v$
14-15	Reference body number where 1 = Sun 2 = Mercury 3 = Venus 4 = Earth 5 = Moon 6 = Mars 7 = Jupiter 8 = Saturn 9 = Uranus 10 = Neptune 11 = Pluto
16-17	Terminating condition number where 1 = reference switch 2 = distance 3 = circular parking orbit 4 = time-of-flight 5 = periapsis 6 = flight path angle 7 = Beta 8 = Burn
18-19	Terminating body number (the numbers used here refer to the same bodies as given in columns 14-15) if $\Delta$ or 0, program sets to 1.
51-70	Identification of subarc

The next card(s) usually contains the C-array data modifications. These cards are not always necessary since the program automatically stores in the C-array all the trajectory information and is capable of starting the new sub-arc without C-array modifications. However, provisions have been made to override the program by the following card(s) of this type:

COLUMN NOS.	DESCRIPTION
1	*indicates the last C-array card, $\Delta$ indicates more C-array cards to follow
2-4	C-array location of variable to be modified (the locations of the variables can be found by checking the Block Data Routine)
5-15	amount of the C-array variable

Four more C-array variables can be placed on a single data card where the variable's field would be 12 columns wide interdispersed with 3 columns used for the C-array storage locations.

Included at the end of the program listings, after the \$DATA card, is a sample setup case for a typical interplanetary trajectory to the planet Jupiter. The case consists of four connected coast arcs: an earth reference coast arc terminating at the earth's sphere-of-influence; a sun reference coast arc terminating at the time,  $t = 8365.78981$  hours; a sun reference coast arc terminating at Jupiter's sphere-of-influence; and finally, a coast arc in Jupiter's reference terminating at periapsis. Notice that for this case some subarcs have C-array modifications.

```

$JOB 1202C003 104 M3RCB M3072B1059
$EXECUTE IBJOB
$IBJOB MAP
$IBFTC TRASUP DECK
1 FORMAT(132H***** GENE
1RAL TRAJECTORY SELECTOR ****
2*****/
379H0*** C ARRAY HAS BEEN INITIALIZED *** REFER TO BLOCK DATA COMPI
4LATION ABOVE ***
5107H0*** CHANGES TO C MAY FOLLOW AS INPUT *** THESE CHANGES REMAIN
6 PERMANENT WITH THIS JOB UNLESS REDEFINED ***
75(1H0/),51H0*** UNITS *** ANGLES ***** = DEGREES/
819X,30HTIME ***** = HOURS/
919X,27HPOSITION W.R.T. PLANET = KM/
119X,27H   "   "   SUN = AU/
219X,31HVELOCITY   "   PLANET = KM/SEC/
319X,31H   "   "   SUN = AU/DAY)
2 FORMAT(80H
1
3 FORMAT(A1,I2,4(1X,A1),4I2,31X,5A6)
4 FORMAT(112X,11HCASE NUMBER I3)
5 FORMAT( 5H0*** ,5A6, 88H***** SUBARC NUMBER I3/
1***** SUBARC NUMBER I3/
21H0 4X,20HMANEUVER TYPE *** = A1,19H (C=COAST , B=BURN)/
3   5X,20HDIRECTION ***** = A1,25H (F=FORWARD , B=BACKWARD)/
4   5X,20HREFERENCE BODY ** = A6/
5   5X,20HTERMINATING COND. = 4A6)
6 FORMAT(1H04X,
1   19HFLIGHT TIME (HRS) =1PE15.7/1H0
2 4X,20HSTATE UPDATE W.R.T. A6/
1 1H013X,10HEPOCH ** =I4,1H/I2,1H/I2,0PF16.2,13H UT (HHMM.SS)/
3   14X,10HPOSITION =1P4E15.7/14X,10HVELOCITY = 4E15.7)
7 FORMAT(67H0***** THIS CASE HAS HAD IT. PROCEED TO NEXT CAS
1E (IF ANY).)
8 FORMAT(A1,79H
1
9 FORMAT(1H0 4X,24HINPUT STATE (X,Y,Z,R) - 2A6/
1 1H013X,10HEPOCH ** =I4,1H/I2,1H/I2,F16.2,13H UT (HHMM.SS)/
1   14X,10HPOSITION =1P4E15.7/
2   14X,10HVELOCITY =4E15.7)
10 FORMAT(1H04X,20HSTATE UPDATE W.R.T. A6 /1H013X,10HPOSITION =
11P4E15.7/14X,10HVELOCITY = 4E15.7)
11 FORMAT(1H04X,9HSTATE OF A6/1H013X,10HT(JULIAN)=1PD15.7/
114X,10HPOSITION =4E15.7/
1   14X,10HVELOCITY =4E15.7)
12 FORMAT(1H04X,20HBURN MANEUVER - TYPE,I2/)
13 FORMAT(14X,10HB C ANGLE=1PE15.7/
1   14X,10HFL PATH = E15.7)
15 FORMAT(14X,10HL AZIMUTH=1PE15.7)
16 FORMAT(14X,10HDELTA HGT=1PE15.7)
17 FORMAT(14X,10HDELTA VEL=1PE15.7)
18 FORMAT(1H0, 4X,19HBETA ***** =1PE15.7)
DOUBLE PRECISION XJD
DIMENSION
1   SUBARC(5),C(1000)  ,POS(4)    ,VEL(4)    ,P(4)      ,V(4)
1   ,G(4)      ,SRV(3)    ,SVV(3)    ,AVEC(4)
COMMON
1   /DATA/C
2   /STATE/ ALPHA,G,DZ,AINV,SQAINV
3   /EPHEM/ XJD,SRV,SVV
EQUIVALENCE

```

```

1      (P,C(413))  ,(V,C(419))  ,(BETA,C(427)) ,(DT,C(428)) ,
2      (POS,C(401)) ,(VEL,C(407)) ,(EPOCH,C(425)),(UT,C(426)) ,
3      (DTBURN,C(435)) ,(BCANG,C(430)) ,(FPANG,C(431)) ,
4      (AZ,C(432)) ,(DELH,C(433)) ,(DELV,C(434)) ,
5      (RTD,C(5)) ,(DTR,C(6))

      INTEGER ERROR
      ERROR = 1
      NOCASE = 0
      WRITE(6,1)
100 READ(5,2)
      WRITE(6,2)
110 NOCASE = NOCASE + 1
      NOARC = 0
      WRITE(6,4) NOCASE
C
C DATA FOR EACH SUBARC OF MISSION
150 READ(5,3) END,MANNO,PUTIN,
      1CDATA,DIREC,TYPE,NOBURN,IBODRF,ITERM,IBODRS,SUBARC
      IREF = 20 * IBODRF + 81
      NOARC = NOARC + 1
      IF(MANNO .NE. 0) NOARC = MANNO
      IF(TYPE .NE. C(322)) GO TO 552
      I = 361
      GO TO 558
152 I = 3 * ITERM + 324
      IRS = 20 * IBODRS + 81
      IF(IBODRS .NE. 0) GO TO 560
      IBODRS = 1
158 IRS = 356
160 WRITE(6,5) SUBARC,NOARC,TYPE,DIREC,C(IREF),C(I),C(I + 1),C(I+2),
      1C(IR)
C
C OVERLAY INPUT TO C ARRAY
      IF( CDATA .EQ. C(324)) CALL NEWC
      IDUM = 357
      IF(PUTIN .NE. C(324)) GO TO 575
C
C INPUT STATE = UPDATED STATE FROM LAST SUBARC
      IDUM = 359
      DO 570 K = 1,3
      POS(K) = P(K)
      570 VEL(K) = V(K)
C
C INITIAL POSITION-VELOCITY PARAMETERS
175 POS(4) = SQRT(POS(1)**2 + POS(2)**2 + POS(3)**2)
      VEL(4) = SQRT(VEL(1)**2 + VEL(2)**2 + VEL(3)**2)
C
C CALENDAR AND JULIAN DATES AT START OF SUBARC
      IF(PUTIN .EQ. C(324)) GO TO 576
      DUM = EPOCH/100.
      IM = (DUM - AINT(DUM)) * 100.
      ID = (EPOCH - AINT(EPOCH)) * 100. + .01
      IY = DUM
      DUM = UT/100.
      XJD = DBLE(DJUL4(IM,ID,IY + 1900)) + DBLE(AINT(DUM + .01)/C(9)
      1   + AINT((DUM - AINT(DUM)) * 100. + .01)/C(8)
      1   + AINT((UT - AINT(UT)) * 100. + .01)/C(7))
176 WRITE(6,9) C(IDUM),C(IDUM + 1),IM,ID,IY,UT,POS,VEL
      CALL ELEM(POS,VEL,IREF,ERROR)
      GO TO (577,2000),ERROR
177 IF(DIREC .NE. C(322)) GO TO 580

```

```

C
C BACKWARD DIRECTION
  DO 578 K = 1,3
  578 VEL(K) = -VEL(K)
C
C MANEUVER TYPE
  580 IF(TYPE .EQ. C(322)) GO TO 700
    DZ = POS(1) * VEL(1) + POS(2) * VEL(2) + POS(3) * VEL(3)
    XMU = C(IREF + 1)
    AINV = 2./POS(4) - VEL(4)**2/XMU
    SQAINV = SQRT(ABS(AINV))
C
C TERMINATING CONDITION
  591 CALL TERM(ITERM,IREF,IBODRS,DIREC,ERROR)
    WRITE(6,18) BETA
    GO TO (600,2000), ERROR
C
C COAST
  600 CALL TWOBOD(IREF,ERROR)
    GO TO (1000,2000), ERROR
C
C BURN
  700 WRITE(6,12) NOBURN
    DT = DTBURN
    IF(NOBURN .NE. 1) GO TO 720
C
C ROTATE
C
C TYPE 1
  715 AVEC(1) = POS(2) * VEL(3) - VEL(2) * POS(3)
    AVEC(2) = VEL(1) * POS(3) - POS(1) * VEL(3)
    AVEC(3) = POS(1) * VEL(2) - VEL(1) * POS(2)
    WRITE(6,13) BCANG,FPANG
    CALL ROTATE (POS,AVEC,BCANG * DTR,P)
    CALL ROTATE (VEL,AVEC,(BCANG - FPANG) * DTR,V)
    GO TO 1000
  720 DO 725 I = 1,4
    P(I) = POS(I)
  725 V(I) = VEL(I)
    GO TO (2000,740,750,760), NOBURN
C
C TYPE 2
  740 WRITE(6,15) AZ
    CALL ROTATE (VEL,POS,-AZ* DTR,V)
    GO TO 1000
C
C STRETCH
C
C TYPE 3 (DELTA H)
  750 WRITE(6,16) DELH
    DELX = (POS(4) + DELH)/POS(4)
    DO 755 I = 1,3
  755 P(I) = DELX * POS(I)
    P(4) = SQRT(P(1)**2 + P(2)**2 + P(3)**2)
    GO TO 1000
C
C TYPE 4 (DELTA V)
  760 WRITE(6,17) DELV
    DELX = (VEL(4) + DELV)/VEL(4)
    DO 765 I = 1,3
  765 V(I) = DELX * V(I)

```

```

      V(4) = SQRT(V(1)**2 + V(2)**2 + V(3)**2)
1000 IDUM = 11
      IF(IBODRF .EQ. 1) IDUM = 9
      DTP = DT * C(IDUM)
C
C UPDATE EPOCH AND JULIAN DATE FOR NEW STATE
      XJD = XJD + DBLE(DTP/C(9))
      CALL JUL4(XJD,IM,ID,IY)
      IY = IY - 1900
      EPOCH = FLOAT(IY) * 100. + FLOAT(IM) + FLOAT(ID)/100. + .005
      DUM = SMGL(XJD - DBLE(DJUL4(IM, ID, IY + 1900))) * C(9)
      UT = AINT(DUM)
      DUM = (DUM - UT) * 60.
      UT = UT * 100. + AINT(DUM) + (DUM - AINT(DUM)) * .6 + .005
      IF(DIREC .NE.C(322)) GO TO 1100
C
C REINSTATE VELOCITY AFTER BACKWARD FLIGHT
      DO 1050 K = 1,3
1050 V(K) = -V(K)
      1100 WRITE(6,6) DTP,C(IREF),
     1                               IM, ID, IY, UT, P, V
      CALL ELEM(P,V,IREF,ERROR)
      GO TO (1101,2000), ERROR
C
C REFERENCE SWITCH (TERM. COND. 1)
      1101 IF(ITERM .NE. 1) GO TO 1200
C
C PLANET STATE AT T ZERO + DT
      IDUM = IBODRF
      IF(IBODRF .EQ. 1) IDUM = IBODRS
      CALL PLANET (IDUM ,ERROR)
      SRVMAG = SQRT(SRV(1)**2 + SRV(2)**2 + SRV(3)**2)
      SVVMAG = SQRT(SVV(1)**2 + SVV(2)**2 + SVV(3)**2)
      WRITE(6,11) C(20 * IDUM + 81),XJD,
     1                               SRV,SRVMAG,SVV,SVVMAG
      GO TO (1130,2000), ERROR
C
C VEHICLE STATE WRT NEW REFERENCE SYSTEM
      1130 IF(IBODRF .EQ. 1) GO TO 1132
C
C PLANET TO SUN UNITS (KM TO AU)
      ONE = 1.
      CONV1 = 1./C(10)
      CONV2 = CONV1 * C(7)
      CONV3 = 1.
      CONV4 = 1.
      GO TO 1134
C
C SUN TO PLANET UNITS (AU TO KM)
      1132 ONE = -1.
      CONV1 = 1.
      CONV2 = 1.
      CONV3 = C(10)
      CONV4 = C(10)/C(7)
      1134 DO 1135 K = 1,3
      P (K) = (P(K) * CONV1 + ONE * SRV(K)) * CONV3
1135 V (K) = (V(K) * CONV2 + ONE * SVV(K)) * CONV4
      P (4) = SQRT(P (1)**2 + P (2)**2 + P (3)**2)
      V (4) = SQRT(V (1)**2 + V (2)**2 + V (3)**2)
      1150 WRITE(6,10) C(IRS),P ,V
      CALL ELEM (P,V,IRS,ERROR)
      GO TO (1200,2000), ERROR
      1200 IF(END .NE. C(320)) GO TO 550
      GO TO 500
      2000 WRITE(6,7)
C
C DUMMY READ TO GET PAST DATA TO NEXT CASE
      2010 READ(5,8) END
      IF(END .NE. C(326)) GO TO 2010
      ERROR = 1
      WRITE(6,8) END
      GO TO 510
      END

```

```

$IBFTC TERM      DECK
      SUBROUTINE TERM(ITERM,IREF,IBODRS,DIREC,ERROR)

C
C  TERMINATING CONDITIONS
C
C  INPUT
C      ITERM = 1  REFERENCE SWITCH
C                  2  SPECIFIED DISTANCE FROM REFERENCE BODY
C                  3  CIRCULAR PARKING ORBIT
C                  4  DELTA TIME
C                  5  PERIAPSIS
C                  6  FLIGHT PATH ANGLE
C                  7  BETA GIVEN
C
C
C  COMMON /DATA/C
1           /STATE/ALPHAW,G,DZ,AINV,SQAINV
2           /EPHEM/ XJD,SRV,SVV
      DIMENSION C(1000),G(4),POS(4),VEL(4),P(4),V(4),BETAK(2),
1             PK(2,4),SRV(3),SVV(3),RPR2(2),ACQVEC(3),DUM(3)
      EQUIVALENCE (POS,C(401)),(VEL,C(407)),(P,C(1)),(BETA,C(427)),
1             (DT,C(428)),(GAMMA,C(429)),(RTD,C(5)),(DTR,C(6)),(TWOP,C(4))
2             ,(P,C(413)),(V,C(419))
      INTEGER ERROR
      DOUBLE PRECISION XJD,TK(2),TEMP
1 FORMAT(59H0** TERM ** SPEAKING. SEMI-MAJOR AXIS IS TOO LARGE. 1/
1A = 1PE15.7)
2 FORMAT(79H0** TERM ** SPEAKING. BETA FAILS TO CONVERGE AFTER 20
1 ITERATIONS. BETA(19) =1PE15.7,13H , BETA(20) =E15.7)
3 FORMAT(90H0** TERM ** SPEAKING. R CANNOT BE REACHED. TERM. CON
1DITION CANNOT BE SATISFIED. 1/A = 1PE15.7,6H , C =E15.7)
4 FORMAT(58H0** TERM ** SPEAKING. ERROR IN PERIAPSIS. I IS NEGATIV
1E.)
5 FORMAT(60H0** TERM ** SPEAKING. FLIGHT PATH ANGLE CANNOT BE OBTAI
1NED.,5H S = 1PE15.7)
6 FORMAT(1H0,4X,19HFLIGHT PATH ANGLE = 1PE15.7,5H DEG.)
7 FORMAT(1H0,4X,19HDISTANCE (KM) *** = 1PE15.7)
8 FORMAT(51H0** TERM ** SPEAKING. PERIAPSIS CANNOT BE REACHED.)
1009 FORMAT(1H04X,19HINFLUENCE RADIUS = 1PE15.7)
1011 FORMAT(105H0** TERM ** SPEAKING. BETA HAS NOT CONVERGED AFTER 20
1 ITERATIONS. VEHICLE MISSED SPHERE OF INFLUENCE BY 1PE15.7,4H AU )
1013 FORMAT(21H0** TERM ** SPEAKING./ 71H PLANET-S SPHERE OF INFLUENCE
1CANNOT BE REACHED. PERIHELION = 1PE15.7,
222H , APHELION = E15.7)
1015 FORMAT(100H* TERM ** SPEAKING. FAILED FIRST GUESS LOGIC, TRANSFER
1MAY MISS PLANETS SPHERE, APPROX CLOSEST DIST=1PE15.7,3H AU)
C
C  BETA GIVEN
      IF(ITERM.EQ. 7) RETURN
      IF(ABS(AINV) .GT. 1.E-20) GO TO 10
      WRITE(6,1) AINV
      9  ERROR = 2
10  XMU = C(IREF + 1)
      SQRTMU = C(IREF + 4)
      DZMU = DZ/SQRTMU
      IF(ITERM .EQ. 4) GO TO 400
C

```

```

C PARAMETERS COMMON TO ALL EXCEPT TERM. COND. NO. 4
  SZ2 = DZ**2 * AINV/XMU
  SZ = DZMU * SQAINV
  CZ = 1. - AINV * POS(4)
  ECC = SQRT(SZ2 + CZ * CZ)
  IF(AINV .LT. 0.) GO TO 20
  EZ = ATAN2(SZ,CZ)
  GO TO 30
20 EZ = ALOG((SZ + CZ)/ECC)
30 GO TO ( 70,200,300,9,500,600 ) + ITERM
C
C REFERENCE SWITCH
70 IF(IREF .NE. 101) GOTO 100
C
C SUN TO PLANET
C
C POSITION OF TRAJECTORY - (SPHERE OF INFLUENCE) INTERCEPTS.
C GROSS ESTIMATE AND INDEPENDENT OF PLANET-S POSITION AT THIS POINT
  TEMP = XJD
  I = 20 * IBODRS + 86
  IGO = 1
  J = I - 2
C
C DETERMINE INNER OR OUTER TRANSFER
  SPHERE = C(J)/C(10)
  RS2 = SPHERE**2
  RAP = C(I)
  RPP = C(I + 1)
  RTARG = (RAP + RPP )/2.
  IF(POS(4) .LT. RTARG) GO TO 72
C
C OUTER TO INNER TRANSFER
  RPER = (1. - ECC)/AINV
  IF(RPER .GT. RAP) GO TO 75
  R1= RPP - SPHERE
  GO TO 77
C
C INNER TO OUTER TRANSFER
72  R1= RAP + SPHERE
  IF(AINV .LT. 0.) GO TO 77
  RAPOG = (1. + ECC)/AINV
  IF(RAPOG .LT. RPP) GO TO 75
  IF(RAPOG .LT. RAP) R1 = RAPOG
  GO TO 77
75  WRITE(6,1013) RPP,RAP
  ERROR = 2
  RETURN
C
C FIRST ACQUISITION VECTOR
  77 R = R1
  GO TO 1201
78  DBETA = BETA/8.
  79 BETAK(IGO) = BETA
  CALL TWOBOD(IREF,ERROR)
  DO 80 K = 1,4
  80 PK(IGO,K) = P(K)
  TK(IGO) = XJD + DBLE(DT)
C
  IF(IGO.EQ.2) GO TO 82
  IGO= IGO + 1
  BETA = BETA- DBETA

```

```

GO TO 79
C
C PLANET POSITIONS AT TIME OF ACQUISITION VECTORS
82 IGO = 1
    DO 85 I = 1,2
    XJD = TK(I)
    CALL PLANET(IBODRS,ERROR)
    IF(ERROR .EQ. 2) RETURN
C
C VEHICLE - PLANET DISTANCES
    DO 84 K = 1,3
    84 DUM(K) = PK(I,K) - SRV(K)
    85 RPR2(I) = DUM(1)**2 + DUM(2)**2 + DUM(3)**2
C
C LOGIC FOR TRAPPING SOLUTION FOR SPHERE OF INFLUENCE
3001 IF(RPR2(2).LT. RPR2(1)) GO TO 3000
    IF(RPR2(1).GT. RS2) GO TO 3003
    IF(RPR2(2).GT.RS2) GO TO 1086
3000 RPR2(1)=RPR2(2)
    BETAK(1)=BETAK(2)
    BETA=BETA-DBETA
    BETAK(2)=BETA
    CALL TWOBO(BREF,ERROR)
    IF(ERROR.EQ.2) RETURN
    XJD=TEMP+DBLE(DT)
    CALL PLANET(IBODRS,ERROR)
    IF (ERROR .EQ.2) RETURN
    DO 3002 K=1,3
    3002 DUM(K)=P(K)-SRV(K)
        RPR2(2)=DUM(1)**2+DUM(2)**2+DUM(3)**2
        GO TO 3001
C END LOGIC
C
C
C TO HANDLE CASE WHEN BETA IS VERY LARGE AND LOCAL MIN HAS BEEN PASSED
3003 BETA=BETAK(1) - DBETA/128.
    CALL TWOBO(BREF,ERROR)
    IF(ERROR.EQ.2) RETURN
    XJD=TEMP+DBLE(DT)
    CALL PLANET(IBODRS,ERROR)
    IF (ERROR .EQ.2) RETURN
    DO 3004 K=1,3
    3004 DUM(K)=P(K)-SRV(K)
        RPR2(2)=DUM(1)**2+DUM(2)**2+DUM(3)**2
        IF( RPR2(2) .GT. RPR2(1) ) GO TO 3005
        DBETA=DBETA/2.
        BETAK(2)=BETA
        GO TO 3000
C
C
C CANNOT OBTAIN FIRST GUESS ,PROBABLY MISSED PLANET
3005 RPR2(1)=SQRT(RPR2(1))
    WRITE(6,1015) RPR2(1)
    ERROR=2
    RETURN
C
C
C BETA ITERATION
1086 DO 86 I = 1,20
    TEST = RPR2(1) - RS2
    BETA = BETAK(1) - TEST * ((BETAK(2) - BETAK(1))/(RPR2(2) - RPR2

```

```

1(1))
IF(ABS(TEST) .LE. 1.E-6) GO TO 87
RPR2(1) = RPR2(2)
BETAK(1) = BETAK(2)
BETAK(2) = BETA
CALL TWOBODY(IREF,ERROR)
IF(ERROR .EQ. 2) RETURN
XJD = TEMP + DBLE(DT)
CALL PLANET(IBODRS,ERROR)
IF(ERROR .EQ. 2) RETURN
DO 1090 K = 1,3
1090 DUM(K) = PI(K) - SRV(K)
86 RPR2(2) = DUM(1)**2 + DUM(2)**2 + DUM(3)**2
C
C END ITERATION
C
C MISSED THE PLANET
RPR2(2) = SQRT(RPR2(2))
WRITE(6,1011) RPR2(2)
ERROR = 2
RETURN
87 XJD = TEMP
RETURN
C
C PLANET TO SUN
100 I = IREF + 3
IGO = 3
GO TO 201
110 WRITE(6,1009) R
190 RETURN
C
C DISTANCE
200 I = IREF + 7
201 R = C(I)
1201 SGN = 1.
IF(DIREC .EQ. C(322)) SGN = -1.
CC = (1. - AINV * R)/ECC
IF((AINV .GT. 0.) .AND. ABS(CC) .GT. 1.) GO TO 210
1205 S = SQRT(ABS(1. - CC**2))
IF(ITERM .EQ. 1) S = SGN * S
IF(AINV .LT. 0.) GO TO 202
EPR = ATAN2(S,CC)
GO TO 204
202 EPR = ALOG(CC + S)
204 SGNE = SIGN(1.,EZ)
IF( SIGN(1.,R - POS(4)) * SGNE .LT. 0.) GO TO 211
E = SGNE * EPR
BETA = (E - EZ)/SQAINV
IF(ITERM .EQ. 1) GO TO 178,78,110,IGO
WRITE(6,7) R
RETURN
210 IF(ABS(CC) .GT. 1.0000005) GO TO 1210
CC = 1.
GO TO 1205
1210 WRITE(6,3) AINV,CC
ERROR = 2
RETURN
211 WRITE(6,4)
ERROR = 2
RETURN
C CIRCULAR PARKING ORBIT

```

```

300 CONTINUE
    RETURN
C
C PERIAPSIS
500 IF((AINV .LE. 0.) .AND. EZ .GT. 0..AND.(C(436).GT.0.)) GO TO 505
    GO TO 510
505 WRITE(6,8)
    ERROR = 2
    RETURN
510 IF( AINV.GT.0.0 .AND. EZ.GT.0.0 ) EZ=EZ-TWOP
    BETA = -EZ/SQAINV
    RETURN
C
C FLIGHT PATH ANGLE
600 GAMRAD= GAMMA * DTR
    WRITE(6,6) GAMMA
    CGAM = COS(GAMRAD)
    S = SQRT(ABS(ECC**2 - 1.)) * SIN(GAMRAD)/(ECC * CGAM)
    IF((ABS(S) .GT. 1.) .OR. ABS(CGAM) .LT. 1.E-20) GO TO 610
    CC = SQRT(1. - SIGN(S**2 , AINV))
    IF(AINV .LT. 0.) GO TO 605
    E = ATAN2(S,CC)
    GO TO 606
605 E = ALOG(S + CC)
606 BETA = (E - EZ)/SQAINV
    RETURN
610 WRITE(6,5) S
    ERROR = 2
    RETURN
C
C DELTA TIME
C
C STARTING VALUE FOR BETA
400 IF(DT .NE. 0.) GO TO 401
    BETA = 0.
    RETURN
401 SQMUDT = SQRTMU * DT
    ECEZ = 1. - AINV * POS(4)
    IF(AINV .GE. 0.) GO TO 410
C
C HYPERBOLIC
    BETA=0.
    GO TO 450
410 IF(ABS(1. - SQRTMU/(SQRT(POS(4)) * VEL(4))) .GT. 1.E-5) GO TO 420
C
C CIRCULAR
    BETA=AINV * SQMUDT
    GO TO 450
C
C ELLIPTIC
420 BETA=(PI - ATAN2(DZMU * SQAINV , ECEZ))/SQAINV
C
C NEWTON-RAPHSON ITERATOR FOR BETA
450 DO 470 K =1,20
    BZ = BETA
    ALPHAW = -AINV * BZ**2
    CALL FALPHA (ERROR)
    IF(ERROR .EQ. 2) RETURN
    BETA = BZ - (POS(4) * G(1) + DZMU * G(2) + G(3) - SQMUDT)/
    1     (DZMU * G(1) + ECEZ * G(2) + POS(4))
470 IF(ABS(1. - BETA/BZ) .LE. 1.E-7) RETURN
    WRITE(6,2) BZ,BETA
    ERROR = 2
    RETURN
    END

```

```

$IBFTC ELEM      DECK
      SUBROUTINE ELEM(POS,VEL,IREF,ERROR)
C
C  ORBITAL ELEMENTS AND STATE PARAMETERS
C
      COMMON /DATA/ C
      1      /STATE/ ALPHA,G,DZ,AINV,SQAINV
      DIMENSION C(1000),POS(4),VEL(4),H(4),P(3),G(4)
      INTEGER ERROR
C
      1 FORMAT(1H013X,10HA ***** =1PE15.7,15H  ECCENTRICITY=E15.7,
     115H  INCLINATION =E15.7/14X,10HARG PER  =E15.7,15H  RA ASC NODE =
     2E15.7,15H  MEAN ANOMALY=A1,E14.7/14X,10HECC. AN. =A1,E14.7,
     315H  TRUE ANOMALY=E15.7,16H  MEAN MOTION = E14.7/
     414X,10HPERIOD  =A1,E14.7,16H  T TO PERIGEE= E14.7,
     415H  ANG MOMENTUM=E15.7/
     514X,10HAPOFOCUS =A1,E14.7,16H  PERIFOCUS   = E14.7)
      2 FORMAT(59H0** ELEM ** SPEAKING. SEMI-MAJOR AXIS IS TOO LARGE. 1/
     1A = 1PE15.7)
C
      FLAG = C(356)
      DZ = POS(1) * VEL(1) + POS(2) * VEL(2) + POS(3) * VEL(3)
      POS(4) = SQRT(POS(1)**2 + POS(2)**2 + POS(3)**2)
      VEL(4) = SQRT(VEL(1)**2 + VEL(2)**2 + VEL(3)**2)
      XMU = C(IREF + 1)
      SQRTMU = C(IREF + 4)
      DZMU = DZ/SQRTMU
      AINV = 2./POS(4) - VEL(4)**2/XMU
      ABAINV = ABS(AINV)
      IF(ABAINV .GT. 1.E-20) GO TO 10
      WRITE(6,2) AINV
      ERROR = 2
      RETURN
      10 SQAINV = SQRT(ABAINV)
      CZ = 1. - AINV * POS(4)
      SZ = DZMU * SQAINV
C
C  SEMI-MAJOR AXIS
      A = 1./AINV
      SZ2 = AINV * DZ**2/XMU
C
C  ECCENTRICITY
      ECC = SQRT(SZ2 + CZ**2)
C
C  ANGULAR MOMENTUM/UNIT MASS
      H(1) = POS(2) * VEL(3) - VEL(2) * POS(3)
      H(2) = VEL(1) * POS(3) - POS(1) * VEL(3)
      H(3) = POS(1) * VEL(2) - VEL(1) * POS(2)
      H(4) = SQRT(H(1)**2 + H(2)**2 + H(3)**2)
      TERMA = (1./POS(4) - AINV)/ECC
      TERM B = DZ/(ECC * XMU)
      DO 40 I = 1,3
      40 P(I) = TERMA * POS(I) - TERM B * VEL(I)
C
C  INCLINATION
      COSI = H(3)/H(4)

```

```

SINI = SQRT(1. - COSI**2)
IF(COSI .NE. 0.) GO TO 45
XI = 90.
GO TO 47
45 XI = ATAN(SINI/COSI) * C(5)
C
C   RIGHT ASCENSION OF ASCENDING NODE
47  CONTINUE
    C(436) = XI
    IF(SINI .EQ. 0.) GO TO 49
    DUM = H(4) * SINI
    SONG = H(1)/DUM
    COMG = -H(2)/DUM
    OMG = ATAN2(SONG,COMG) * C(5)
C
C   ARGUMENT OF PERIGEE
    ARGPER = ATAN2(P(3)/SINI,P(1) * COMG + P(2) * SONG) * C(5)
49  CONTINUE
C
C   MEAN MOTION
48 XMOT = SQRTMU * ABAINV * SQAINV
C
C   MEAN AND ECCENTRIC ANOMALY
    IF( A.LT.0. ) GO TO 200
    EZ = ATAN2(SZ,CZ)
    GO TO 201
200 EZ=ALOG1 (CZ+SZ)/ECC )
201 XMA = EZ - SZ
    IF( A.LT.0. ) XMA=-XMA
C
C   TRUE ANOMALY
    E2=EZ/2.
    DUM= (1.+ECC)/(1.-ECC)
    IF(A.LT.0.) GO TO 205
    Y=SQRT(DUM)*SIN(E2)
    X=COS(E2)
    GO TO 206
205 EX1=EXP(E2)
    EX2=EXP(-E2)
    Y=SQRT(-DUM)*.5*(EX1-EX2)
    X=.5*(EX1+EX2)
206 TA=2.*ATAN2(Y,X)*C(5)
C
C   TIME FROM PERIGEE
30 TPER = -XMA/XMOT
C
C   FLAG IRRELAVENT PARAMETERS FOR HYPERBOLIC ORBITS
    IF(A .LT. 0.) FLAG=C(320)
C
C   BYPASS PERIOD AND APOFOCUS CALC IF HYPERBOLIC
    IF( A.LT.0. ) GO TO 202
C
C   PERIOD
    PERIOD = C(4) *      A **1.5/SQRTMU
C
C   APOFOCUS
    RA = A * (1. + ECC)
C
C   PERIFOCUS
202 DUM = 1. - ECC
    IF(ABS(DUM) .GT. 1.E-3) GO TO 60
    ALPHA = -EZ**2 * SIGN(1.+A)
    BETA = -EZ/SQAINV
    CALL FALPHA(ERROR)
    IF(ERROR .EQ. 2) RETURN
50  RP = G(2) + POS(4) * G(4) + DZMU * G(1)
    GO TO 70
60  RP = A * DUM
C
70  IF( A.LT.0. ) GO TO 100
    EZ = EZ * C(5)
    XMA = XMA * C(5)
100 WRITE(6,1) A,ECC,XI,ARGPER,OMG,FLAG,XMA,FLAG,EZ,TA,XMOT,
    1FLAG,PERIOD,      TPER,H(4),FLAG,RA,RP
    RETURN
    END

```

```

$IBFTC BLOCK    DECK
C
C  CONSTANT AND DATA DEFINITIONS FOR GENERAL TRAJECTORY SELECTOR -
C  SUBJECT TO OVERLAY VIA INPUT.
C
C      BLOCK DATA
C      COMMON /DATA/C
C      DIMENSION C(1000)
C*****
C  PERM. PROGRAM CONSTANTS
    DATA(C(I) , I = 1,100)
    1/3.14159265      ,1.57079633   ,4.71238899   ,6.28318531   , C     1
    1 57.2957795      ,1.74532925E-2 ,86400.        ,1440.        , C     5
    1 24.             ,149599000.     ,.277777778 E-3,89 * 0     , C     9
C*****
C  BODY CONSTANTS - (**1) = BODY NAME
C                  (**2) = GRAVITATIONAL CONSTANT (KM**3/SEC**2)
C                  (**3) = BODY RADIUS (KM)
C                  (**4) = RADIUS OF SPHERE OF INFLUENCE (KM)
C                  (**5) = SQRT(GRAV. CONST.)
C                  (**6) = APHELION DISTANCE (AU)
C                  (**7) = PERIHELION DISTANCE (AU)
C                  (**8) = TERMINATING DISTANCE (TERM. COND. NO. 2)
C
C  CONVERSION  ER**3/HR**2 TO KM**3/SEC**2 MULT. BY 2.0020838E4
C  ER**3/HR**2 TO AU**3/DAY**2 MULT. BY 4.463992E-11
C
C  SUN - C(102) = AU**3/DAY**2 , ALL DISTANCES = AU
    DATA(C(I) , I = 101,160)
    1/6HSUN            ,2.9591203 E-4,10000.        ,1.E10      , C     101
    2 1.7202094       E-2,0.          ,0.            ,1.E5      , C     105
    312 * 0.           ,
C
C  MERCURY
    1 6HMERC           ,19 * 0.          ,               , C     121
C
C  VENUS
    1 6HVENUS          ,324769.5     ,0.            ,523647.35   , C     141
    2 569.885515       ,.72824327   ,.71842093   ,1.E3      , C     145
    312 * 0./
    DATA (C(I) , I = 161,220)
C
C  EARTH
    1/6HEARTH          ,398603.2     ,6378.165     ,786427.74   , C     161
    2 631.3503         ,1.01672794   ,.98327228   ,1.E3      , C     165
    312 * 0.           ,
C
C  MOON
    1 6HMOON           ,4900.7588   ,1738.        ,66000.        , C     181
    2 170.00542        ,32768.        ,1.            ,1.E3      , C     185
    312 * 0.           ,
C
C  MARS
    1 6HMARS           ,42977.795   ,0.            ,464968.23   , C     201
    2 207.310863       ,1.66605432   ,1.38132232   ,1.E3      , C     205
    312 * 0./
    DATA (C(I) , I = 221,280)
C
C  JUPITER

```

```

1/6HJUPITER      ,1.2671077E8   ,0.          ,4.8E7      , C  221
2 1.1256588 E4   ,5.45381528    ,4.95261347    ,1.E3      , C  225
312 * 0.          ,
C
C  SATURN
1 6HSATURN       ,19 * 0.        ,
C
C  URANUS
1 6HURANUS       ,19 * 0.        ,
DATA (C(I) + I = 281,319)
C
C  NEPTUNE
1/6HNEPTUN        ,19 * 0.        ,
C
C  PLUTO
1 6HPLUTO         ,18 * 0.        ,
/          C  281
C
***** DATA(C(I) + I = 320,363)
1/1H*              ,1HC          ,1HB          ,1HF      , C  320
2 1HY              ,1HN          ,1H1          ,6HREF. S  , C  324
3 6HWITCH          ,6HINTO       ,6HSPECIF    ,6HIED DI  , C  328
4 6HSTANCE         ,6HCIRCUL   ,6HAR PAR    ,6HK ORB.  , C  332
5 6HDELTA          ,6HTIME      ,6H          ,6HPERIAP  , C  336
6 6HSIS            ,6H          ,6HFLIGHT    ,6H PATH   , C  340
7 6HANGLE          ,6HBETA G   ,6HIVEN      ,6H          , C  344
8 6H              ,6H          ,6H          ,6H          , C  348
9 6H              ,6H          ,6H          ,6H          , C  352
1 6H              ,6HINITIA   ,6HL          ,6HUPDATE  , C  356
1 6HD              ,6HBURN M   ,6HANEUVE   ,6HR ONLY   / C  360
*****
C
C  DATA (C(I), I = 401,446)
C
X INPUT           Y INPUT        Z INPUT        R INPUT      , C  401
1/1.2777285E3    ,6.0337409E3  ,2.2448028E3  ,6.5633649E3
C
*****          *****          XDOT INPUT    YDOT INPUT      , C  405
2 0.              ,0.          ,-1.4655730E1 ,2.2651264
C
ZDOT INPUT        RDOT INPUT     *****        *****      , C  409
3 2.2535885     ,1.5E1        ,0.          ,0.          ,
C
X OUTPUT          Y OUTPUT        Z OUTPUT        R OUTPUT      , C  413
4 -7.884552E5   ,-1.5145603E5 ,1.4159293E4 ,8.0299532E5
C
*****          *****          XDOT OUTPUT  YDOT OUTPUT      , C  417
5 0.              ,0.          ,-1.0017459E1 ,2.0400995
C
ZDOT OUTPUT        RDOT OUTPUT     *****        *****      , C  421
6 1.3451798E-1  ,1.0223971E1  ,0.          ,0.          ,
C
YYMM.DD           HHMM.SSSS      BETA          TIME OF FLIGHT , C  425
7 6912.30         ,2307.5100   ,0.          ,0.          ,
C
FLIGHT PATH ANGLE H(X)          H(Y)          H(Z)      , C  429
8 0.              ,0.          ,0.          ,0.          ,
C
ANG MOMENTUM      SEMIMAJOR AXIS ECCENTRICITY INCLINATION
9 0.              ,0.          ,0.          ,0.          , C  433
C
ANG OF             ANG OF        MEAN          ECCENTRIC

```

```

C      NODE          PERICENTER        MOTION        ANOMOLY      C  437
C      1 0.           ,0.             ,0.           ,0.           C
C      MEAN          TRUE            TIME          C
C      ANOMOLY       ANOMOLY        FROM PERIGEE   PERIOD      C  441
C      2 0.           ,0.             ,0.           ,0.           C
C      APOFOCUS      PERIFOCUS      /             C  445
C      3 0.           ,0.             /
C ****EPHEMERIS PARAMETERS FOR PLANETARY BODIES
C DATA(C(I) , I = 501,536)
C
C SUN (1)
C L = 1 *** 2439500.5 TO 2441324.5
C A(1),A(2),A(3)
C
C      1    /2.0973284E-01,-8.9704818E-01,-3.8899223E-01,      C  501
C
C      B(1),B(2),B(3)
C
C      1    9.7762109E-01, 1.9239543E-01, 8.3423951E-02,      C  504
C
C      MEAN MOTION,MEAN ANOMALY,ECCENTRICITY
C
C      1    1.7202146E-02, 3.2352436E 00, 1.6724545E-02,      C  507
C
C      L = 2 *** 2441324.5 TO 2443148.5
C
C      1    2.1151123E-01,-8.9670185E-01,-3.8883106E-01,      C  510
C      1    9.7723923E-01, 1.9402791E-01, 8.4128786E-02,      C  513
C      1    1.7202114E-02, 3.1941870E 00, 1.6727522E-02,      C  516
C
C      L = 3 *** 2443148.5 TO 2444972.5
C
C      1    2.1012915E-01,-8.9697996E-01,-3.8893901E-01,      C  519
C      1    9.7753770E-01, 1.9276143E-01, 8.3576151E-02,      C  522
C      1    1.7202113E-02, 3.1563250E 00, 1.6702466E-02,      C  525
C
C      L = 4 *** 2444972.5 TO 2446796.5
C
C      1    2.1110485E-01,-8.9679081E-01,-3.8884567E-01,      C  528
C      1    9.7732664E-01, 1.9365813E-01, 8.3960217E-02,      C  531
C      1    1.7202124E-02, 3.1160858E 00, 1.6727726E-02/      C  534
C MERCURY (2)
C DATA(C(I) , I = 537,572)
C
C      1    /-0.           ,--0.          ,--0.          ,      C  537
C      1    -0.           ,--0.          ,--0.          ,      C  540
C      1    -0.           ,--0.          ,--0.          ,      C  543
C      1    -0.           ,--0.          ,--0.          ,      C  546
C      1    -0.           ,--0.          ,--0.          ,      C  549
C      1    -0.           ,--0.          ,--0.          ,      C  552
C      1    -0.           ,--0.          ,--0.          ,      C  555
C      1    -0.           ,--0.          ,--0.          ,      C  558
C      1    -0.           ,--0.          ,--0.          ,      C  561
C      1    -0.           ,--0.          ,--0.          ,      C  564
C      1    -0.           ,--0.          ,--0.          ,      C  567
C      1    -0.           ,--0.          ,--0.          /      C  570
C VENUS (3).

```

```

      DATA(C(I) , I = 573+608)
1   /-4.7206284E-01, 4.8788472E-01, 2.4966932E-01,          C  573
1   -5.4646455E-01,-4.4401339E-01,-1.6557205E-01,          C  576
1   2.7962447E-02,-2.5356522E 00, 6.7856914E-03,          C  579
1   -4.7277628E-01, 4.8730243E-01, 2.4945720E-01,          C  582
1   -5.4584799E-01,-4.4464735E-01,-1.6590523E-01,          C  585
1   2.7962433E-02,-1.7989062E 00, 6.7883050E-03,          C  588
1   -4.7245589E-01, 4.8755941E-01, 2.4955992E-01,          C  591
1   -5.4612458E-01,-4.4435794E-01,-1.6576735E-01,          C  594
1   2.7962475E-02,-1.0602999E 00, 6.7817544E-03,          C  597
1   -4.7150609E-01, 4.8832905E-01, 2.4985205E-01,          C  600
1   -5.4694541E-01,-4.4350646E-01,-1.6534188E-01,          C  603
1   2.7962459E-02,-3.2052571E-01, 6.7896492E-03/          C  606

C EARTH (4)
      DATA(C(I) , I = 609+644)
1   /-2.0973284E-01, 8.9704818E-01, 3.8899223E-01,          C  609
1   -9.7762109E-01,-1.9239543E-01,-8.3423951E-02,          C  612
1   1.7202146E-02, 3.2352436E 00, 1.6724545E-02,          C  615
1   -2.1151123E-01, 8.9670185E-01, 3.8883106E-01,          C  618
1   -9.7723923E-01,-1.9402791E-01,-8.4128786E-02,          C  621
1   1.7202114E-02, 3.1941870E 00, 1.6727522E-02,          C  624
1   -2.1012915E-01, 8.9697996E-01, 3.8893901E-01,          C  627
1   -9.7753770E-01,-1.9276143E-01,-8.3576151E-02,          C  630
1   1.7202113E-02, 3.1563250E 00, 1.6702466E-02,          C  633
1   -2.1110485E-01, 8.9679081E-01, 3.8884567E-01,          C  636
1   -9.7732664E-01,-1.9365813E-01,-8.3960217E-02,          C  639
1   1.7202124E-02, 3.1160858E 00, 1.6727726E-02/          C  642

C MOON (5)
      DATA(C(I) , I = 645+680)
1   /-0.          ,-0.          ,-0.          ,          C  645
1   -0.          ,-0.          ,-0.          ,          C  648
1   -0.          ,-0.          ,-0.          ,          C  651
1   -0.          ,-0.          ,-0.          ,          C  654
1   -0.          ,-0.          ,-0.          ,          C  657
1   -0.          ,-0.          ,-0.          ,          C  660
1   -0.          ,-0.          ,-0.          ,          C  663
1   -0.          ,-0.          ,-0.          ,          C  666
1   -0.          ,-0.          ,-0.          ,          C  669
1   -0.          ,-0.          ,-0.          ,          C  672
1   -0.          ,-0.          ,-0.          ,          C  675
1   -0.          ,-0.          ,-0.          /          C  678

C MARS (6)
      DATA(C(I) , I = 681+716)
1   /1.3829985E 00,-5.6630965E-01,-2.9706821E-01,          C  681
1   6.3562427E-01, 1.2582528E 00, 5.6049994E-01,          C  684
1   9.1460837E-03,-9.4738666E-01, 9.3422198E-02,          C  687
1   1.3832179E 00,-5.6588296E-01,-2.9686122E-01,          C  690
1   6.3515644E-01, 1.2584498E 00, 5.6061370E-01,          C  693
1   9.1460814E-03, 3.1682745E 00, 9.3357480E-02,          C  696
1   1.3832166E 00,-5.6588123E-01,-2.9684368E-01,          C  699
1   6.3514318E-01, 1.2584350E 00, 5.6061798E-01,          C  702
1   9.1461283E-03, 1.0012256E 00, 9.3435113E-02,          C  705
1   1.3836980E 00,-5.6493266E-01,-2.9640773E-01,          C  708
1   6.3410942E-01, 1.2588681E 00, 5.6084985E-01,          C  711
1   9.1461267E-03,-1.1665322E 00, 9.3348135E-02/          C  714

C JUPITER (7)
      DATA(C(I) , I = 717+752)
1   /5.0598305E 00, 1.1549097E 00, 3.7173985E-01,          C  717
1   -1.2062181E 00, 4.6350877E 00, 2.0179477E 00,          C  720
1   1.4500298E-03, 3.0606442E 00, 4.8153945E-02,          C  723
1   5.0553738E 00, 1.1694208E 00, 3.7807367E-01,          C  726

```

```

1   -1.2220346E 00, 4.6308802E 00, 2.0165227E 00, C 729
1   1.4502952E-03, 5.7022524E 00, 4.8054559E-02, C 732
1   5.0468548E 00, 1.2059230E 00, 3.9394118E-01, C 735
1   -1.2617882E 00, 4.6227371E 00, 2.0139991E 00, C 738
1   1.4497771E-03, 2.0567483E 00, 4.7919003E-02, C 741
1   5.0266418E 00, 1.2735259E 00, 4.2342471E-01, C 744
1   -1.3354298E 00, 4.6035460E 00, 2.0074230E 00, C 747
1   1.4502550E-03, 4.6855981E 00, 4.8063546E-02/ C 750
C  SATURN (8)
    DATA(C(I) , I = 753,788)
1   /-5.2867721E-01, 8.7798163E 00, 3.6530586E 00, C 753
1   -9.4879417E 00,-6.3309856E-01, 1.4848665E-01, C 756
1   7.3981952E-04, 5.2957826E 00, 5.4112226E-02, C 759
1   -2.4626776E-01, 8.8080769E 00, 3.6527202E 00, C 762
1   -9.5134958E 00,-3.7277490E-01, 2.5749649E-01, C 765
1   7.3813744E-04, 1.0623819E-01, 5.4813042E-02, C 768
1   -7.0073493E-01, 8.8089353E 00, 3.6730199E 00, C 771
1   -9.5213411E 00,-7.9237206E-01, 8.3859550E-02, C 774
1   7.3454244E-04, 1.1241490E 00, 5.6139224E-02, C 777
1   -4.9209815E-01, 8.8163435E 00, 3.6672647E 00, C 780
1   -9.5284843E 00,-5.9998196E-01, 1.6379992E-01, C 783
1   7.3550969E-04, 2.2172488E 00, 5.1246147E-02/ C 786
C  URANUS (9)
    DATA(C(I) , I = 789,824)
1   /-1.9032647E 01, 2.6363596E 00, 1.4240991E 00, C 789
1   -2.9821565E 00,-1.7427136E 01,-7.5936648E 00, C 792
1   2.0340827E-04, 1.8942232E 00, 5.0800421E-02, C 795
1   -1.8879279E 01, 2.9112393E 00, 1.5424296E 00, C 798
1   -3.2817947E 00,-1.7292450E 01,-7.5306261E 00, C 801
1   2.0504200E-04, 5.8105500E-01, 4.5854992E-02, C 804
1   -1.8978032E 01, 2.7129594E 00, 1.4564429E 00, C 807
1   -3.0653878E 00,-1.7378891E 01,-7.5710517E 00, C 810
1   2.0405711E-04, 9.3773049E-01, 4.9448118E-02, C 813
1   -1.9241125E 01, 1.1346875E 00, 7.6939741E-01, C 816
1   -1.3464271E 00,-1.7612001E 01,-7.6977830E 00, C 819
1   2.0304725E-04, 1.2254305E 00, 4.7767922E-02/ C 822
C  NEPTUNE (10)
    DATA(C(I) , I = 825,860)
1   /1.9291692E 01, 2.1619241E 01, 8.3703197E 00, C 825
1   -2.3172047E 01, 1.7651633E 01, 7.8148856E 00, C 828
1   1.0385988E-04, 3.2680243E 00, 5.4339129E-03, C 831
1   2.4779832E 01, 1.5990521E 01, 5.9272870E 00, C 834
1   -1.7038688E 01, 2.2785694E 01, 9.7617503E 00, C 837
1   1.0426883E-04, 3.7382858E 00, 8.9812170E-03, C 840
1   1.5534306E 01, 2.3931650E 01, 9.4122013E 00, C 843
1   -2.5705566E 01, 1.4151179E 01, 6.4445135E 00, C 846
1   1.0446296E-04, 3.4966100E 00, 8.4422911E-03, C 849
1   3.0246848E 01, 5.6000010E-01,-5.3173721E-01, C 852
1   -3.1654692E-01, 2.7993687E 01, 1.1475445E 01, C 855
1   1.0336214E-04, 4.7080778E 00, 6.2333927E-03/ C 858
C  PLUTO (11)
    DATA(C(I) , I = 861,896)
1   /-2.7648371E 01,-2.8510578E 01,-6.3179730E-01, C 861
1   2.5446812E 01,-2.4335954E 01,-1.5401763E 01, C 864
1   6.8717631E-05, 5.7817696E 00, 2.5263070E-01, C 867
1   -2.6905922E 01,-2.8729446E 01,-9.1503799E-01, C 870
1   2.5758069E 01,-2.3636926E 01,-1.5265829E 01, C 873
1   6.9631164E-05, 5.8904512E 00, 2.4745160E-01, C 876
1   -2.7673876E 01,-2.8463870E 01,-6.1516616E-01, C 879
1   2.5398347E 01,-2.4360391E 01,-1.5409970E 01, C 882
1   6.8759175E-05, 6.0327829E 00, 2.5241916E-01, C 885
1   -2.7179941E 01,-2.8928263E 01,-8.9727677E-01, C 888
1   2.5892389E 01,-2.3851123E 01,-1.5359832E 01, C 891
1   6.8759669E-05, 6.1478175E 00, 2.5334895E-01/ C 894
*****END*****

```

```

SIBFTC DJUL4 DECK,LIST,M94,XR7
CDJUL4
      FUNCTION DJUL4(NM,ND,NY)
C
C      VERSION OF 09/30/64
C      FORTRAN FUNCTION
C      FOR USE WITH FORTRAN 4 ON IBM 7094
C
C      PURPOSE
C          COMPUTES JULIAN DATE AT 0 HOURS UNIVERSAL TIME (OR
C          0 HOURS EPHemeris TIME).
C
C      CALLING SEQUENCE
C          NAME =DJUL4(NM,ND,NY)
C
C      INPUT
C          NM    = CALENDAR MONTH
C          ND    = CALENDAR DAY
C          NY    = CALENDAR YEAR
C
C      OUTPUT
C          NAME = JULIAN DATE AT 0 HOURS UNIVERSAL TIME
C
C      REFERENCE
C          REFER TO MATHEMATICAL DESCRIPTION IN SUBPROGRAM WRITEUP
C
C      METHOD
C          THE NUMBER OF DAYS WHICH HAVE ELAPSED FROM 12 HOURS
C          UNIVERSAL TIME JAN. 0, 1800 ARE COUNTED AND ADDED TO THE
C          JULIAN DATE OF 12 HOURS UNIVERSAL TIME OF JAN. 0, 1800.
C
C      RESTRICTIONS
C          DATE RESTRICTED TO LIE BETWEEN JANUARY 1, 1801 AND DECEMBER
C          31, 2000.
C
C      ACCURACY
C          EXACT BINARY REPRESENTATION WITHIN DATE LIMITATIONS.
C
C      REQUIRED SUBPROGRAMS - FORTRAN 4 LIBRARY
C          NONE
C
C      REQUIRED SUBPROGRAMS - OTHER
C          NONE
C
C      STORAGE REQUIREMENTS
C
C      TIMING
C          NO ESTIMATE AVAILABLE
C
C      ANALYSIS
C          RICHARD J. SANDIFER, DATA SYSTEMS DIVISION
C          GODDARD SPACE FLIGHT CENTER, NASA
C
C      PROGRAMMER
C          RICHARD J. SANDIFER, DATA SYSTEMS DIVISION
C          ROBERT A. DEVANEY, NASA, THEORETICAL DIVISION
C
C      PROGRAM MODIFICATIONS
C          07/22/63 ORIGINAL BY R. J. SANDIFER
C          09/30/64 MODIFIED BY R.A. DEVANEY TO FORTRAN 4
C
C***** START PROGRAM *****
C
C      2 DIMENSION RM(12)
C      DATA (RM(I),I=1,12)/0.,31.,28.,31.,30.,31.,30.,31.,
C      1      30./
C      17 Y=NY-1800
C      18 YL=AINT((Y-1.0)/4.0)
C      19 YC=AINT((Y+99.0)/100.0)-1.0
C      20 RY=Y-YL
C      21 DJUL4 = RY*365.0+YL*366.0-YC+2378495.5
C      22 TD=ND
C      23 DO 24 N=1,NM
C      24 DJUL4 = DJUL4 + RM(N)
C      25 IF(NM .LE. 2) GO TO 29
C      26 IF(Y .EQ. 100.0) GO TO 29
C      27 IF ( MOD (NY,4)) 29,28,29
C      28 DJUL4 = DJUL4 + 1.
C      29 DJUL4 = DJUL4 + TD
C      RETURN
C      END
C
C      DJUL 004
C      DJUL 006
C      DJUL 008
C      DJUL 009
C      DJUL 010
C      DJUL 011
C      DJUL 012
C      DJUL 013
C      DJUL 015
C      DJUL 016
C      DJUL 017
C      DJUL 018
C      DJUL 019
C      DJUL 020
C      DJUL 021
C      DJUL 022
C      DJUL 023
C      DJUL 024
C      DJUL 025
C      DJUL 026
C      DJUL 027
C      DJUL 028
C      DJUL 029
C      DJUL 030
C      DJUL 031
C      DJUL 032
C      DJUL 033
C      DJUL 034
C      DJUL 035
C      DJUL 036
C      DJUL 037
C      DJUL 038
C      DJUL 040
C      DJUL 041
C      DJUL 042
C      DJUL 043
C      DJUL 044
C      DJUL 045
C      DJUL 047
C      DJUL 048
C      DJUL 049
C      DJUL 050
C      DJUL 051
C      DJUL 052
C      DJUL 053
C      DJUL 054
C      DJUL 055
C      DJUL 056
C      DJUL 057
C      DJUL 058
C      DJUL 059
C      DJUL 061
C      DJUL 062
C      DJUL 063
C      DJUL 064
C      DJUL 079
C      DJUL 082
C      DJUL 084
C      DJUL 085
C      DJUL 092
C      DJUL 093

```

```

SIBFTC JUL4      DECK,LIST,M94,XR7
CJUL4
      SUBROUTINE JUL4  (DJ,NM,ND,NY)                                JULCAL04
C
C      VERSION OF 09/30/64                                         JULCAL06
C          FORTRAN SUBROUTINE
C          FOR USE WITH FORTRAN 4 ON IBM 7094
C
C      PURPOSE
C          COMPUTES CALENDAR DATE FROM JULIAN DATE AT 0 HOURS      JULCAL08
C          UNIVERSAL TIME (OR 0 HOURS EPHemeris TIME).                JULCAL09
C
C      CALLING SEQUENCE
C          CALL JUL4  (DJ,NM,ND,NY)                                JULCAL10
C
C      INPUT
C          DJ    = JULIAN DATE AT 0 HOURS UNIVERSAL TIME            JULCAL12
C
C      OUTPUT
C          NM    = CALENDAR MONTH                                     JULCAL13
C          ND    = CALENDAR DAY                                      JULCAL15
C          NY    = CALENDAR YEAR                                     JULCAL16
C
C      REFERENCE
C          REFER TO MATHEMATICAL DESCRIPTION IN SUBPROGRAM WRITEUP JULCAL17
C
C      METHOD
C          THE NUMBER OF DAYS FROM 12 HOURS UNIVERSAL TIME JAN. 0, JULCAL18
C          1800 IS CALCULATED. THE INTEGRAL NUMBER OF YEARS IN THIS JULCAL19
C          NUMBER IS ADDED TO 1800 TO GIVE THE CURRENT CALENDAR YEAR JULCAL20
C          AND THE NUMBER OF DAYS CONTAINED IN THE INTEGRAL NUMBER JULCAL21
C          OF YEARS ELAPSED SINCE JAN. 0, 1800 IS SUBTRACTED FROM THE JULCAL22
C          ORIGINAL NUMBER OF DAYS FROM JAN. 0, 1800. THE INTEGRAL JULCAL23
C          NUMBER OF MONTHS IN THIS REMAINDER IS CALCULATED TO GIVE JULCAL24
C          THE CURRENT CALENDAR MONTH. THE NUMBER OF DAYS CONTAINED JULCAL25
C          IN THIS INTEGRAL NUMBER OF MONTHS IS SUBTRACTED FROM THE JULCAL26
C          PREVIOUS REMAINING DAYS TO GIVE THE CALENDAR DAY.          JULCAL27
C          APPROPRIATE CONSIDERATION HAS BEEN GIVEN TO THOSE YEARS JULCAL28
C          WHICH ARE DIVISIBLE BY 4, 100, AND 400.                  JULCAL29
C
C      RESTRICTIONS
C          DATE RESTRICTED TO LIE BETWEEN JANUARY 1, 1801 AND DECEMBER JULCAL30
C          31, 2000.                                                 JULCAL31
C
C      ACCURACY
C          *****
C
C      REQUIRED SUBPROGRAMS - FORTRAN 4 LIBRARY
C          NONE                                                 JULCAL32
C
C      REQUIRED SUBPROGRAMS - OTHER
C          09/30/64  DJUL4                                         JULCAL33
C
C      STORAGE REQUIREMENTS
C
C      TIMING

```

```

C      NO ESTIMATE AVAILABLE          JULCAL58
C
C      ANALYSIS                      JULCAL59
C          RICHARD J. SANDIFER, DATA SYSTEMS DIVISION    JULCAL60
C          GODDARD SPACE FLIGHT CENTER, NASA             JULCAL61
C
C      PROGRAMMER                     JULCAL62
C          RICHARD J. SANDIFER, DATA SYSTEMS DIVISION    JULCAL63
C          ROBERT A. DEVANEY , NASA , THEORETICAL DIVISION JULCAL64
C
C      PROGRAM MODIFICATIONS        JULCAL65
C          07/22/63 ORIGINAL BY R. J. SANDIFER           JULCAL66
C          09/30/64 MODIFIED BY R.A. DEVANEY TO FORTRAN 4  JULCAL67
C
C***** START PROGRAM *****          JULCAL68
C
C      2 DIMENSION M(12)              JULCAL69
C      DATA (M(I),I=1,12)/31+28,31+30,31+30+31+31,30+31+30+31/
C      6 NM=0                          JULCAL70
C          M(2) = 28                  JULCAL71
C
C      20 IF(DJ .GT. 2415020.5) GO TO 22
C      21 NY= INT ((DJ-2378495.51/365.25)+1800          JULCAL72
C          GO TO 23                  JULCAL73
C      22 NY= INT ((DJ-2378495.5+1.0)/365.25)+1800
C      23 NDY=DJ-DJUL4(1,0,NY)          JULCAL74
C      24 ND=NDY                      JULCAL75
C      25 DO 33 N=1,12                JULCAL76
C      26 IF(N .NE. 2) GO TO 30
C      27 IF(NY .EQ. 1900) GO TO 30
C      28 IF ( MOD (NY,4)) 30+29,30
C      29 M(N)=M(N)+1                JULCAL77
C      30 ND=ND-M(N)                 JULCAL78
C      31 NM=NM+1                   JULCAL79
C      32 IF (ND) 34,34,33          JULCAL80
C      33 CONTINUE                   JULCAL81
C      34 ND=ND+M(N)                 JULCAL82
C      RETURN
C      END

```

```

$IBFTC NEWC      LIST,DECK
C
C  READ DATA THAT OVERLAYS THE C ARRAY
C  5 ITEMS OR LESS PER CARD
C  AN * IN COLUMN 1 INDICATES LAST CARD
    SUBROUTINE NEWC
1  FORMAT(A1,I3,E11.4,4(I3,E12.5))
2  FORMAT(1HO 4X,28HOVERLAY INPUT TO C ARRAY ***/
11H0, 5X,103HI      C (I)          I      C (I)           I      C
2(I)                 I      C (I)          I      C (I))
3  FORMAT(      5(I8,1PE15.7))
    COMMON
1    /DATA/C
    DIMENSION
1    C(1000)  ,BUFFER(5),INDEX(5)
    WRITE(6,2)
100 READ(5,1) END,((INDEX(I),BUFFER(I)),I=1,5)
    WRITE(6,3)   ((INDEX(I),BUFFER(I)),I=1,5)
    DO 200 I = 1,5
    K = INDEX(I)
    IF(K .EQ. 0) GO TO 200
    C(K) = BUFFER(I)
200 CONTINUE
    IF(END .NE. C(320))GO TO 100
    RETURN
    END

```

```

$IBFTC TWOBOD LIST,DECK
      SUBROUTINE TWOBOD(IREF,ERROR)
C
C   TWO-BODY
C
C   INPUT - POS - POSITION VECTOR (4) AT LAST RECTIFICATION (X,Y,Z,R)
C           VEL - VELOCITY VECTOR (4) AT LAST RECTIFICATION (X,Y,Z,R)
C           XMU - MU
C           SQRTMU - SQRT(MU)
C           BETA
C
C   OUTPUT  P - POSITION VECTOR UPDATED
C           V - VELOCITY VECTOR UPDATED
C           DZ (POS . VEL)
C           AINV (1./A)
C
C   DIMENSION C(1000),G(4),      P(4),V(4)
C   1,POS(4),VEL(4)
C   COMMON /DATA/C
C   1          /STATE/ALPHA,G,DZ,AINV,SQAINV
C   EQUIVALENCE (POS,C( 401)),(VEL,C( 407)),(BETA,C( 427)),
C   1(DT,C(428)),(P,C(413)),(V,C(419))
C   INTEGER ERROR
C   XMU = C(IREF + 1)
C   SQRTMU = C(IREF + 4)
C   DZMU = DZ/SQRTMU
C   ALPHA = -AINV * BETA**2
C   CALL FALPHA (ERROR)
C   IF(ERROR .EQ. 2) RETURN
100  P(4) = POS(4) * G(4)+DZMU * G(1) + G(2)
      DT = (POS(4) * G(1) + DZMU * G(2) + G(3))/SQRTMU
      FS = 1. - G(2)/POS(4)
      GS=POS(4)*G(1)/SQRTMU + DZ * G(2)/XMU
      FDOT = -SQRTMU * G(1)/(POS(4) * P(4))
      GDOT = 1. - G(2)/P(4)
      DO 200 I = 1,3
      P(I) = FS * POS(I) + GS * VEL(I)
200  V(I) = FDOT * POS(I) + GDOT * VEL(I)
      V(4) = SQRT(V(1)**2 + V(2)**2 + V(3)**2)
      RETURN
      END

```

```

$IBFTC FALPHA LIST,DECK
C
C INPUT ALPHA
C      BETA
C
C OUTPUT G(I) , I=1,4
      SUBROUTINE FALPHA (ERROR)
      COMMON /STATE/ ALPHA,G,DZ,AINV,SQAINV
      1       /DATA/
      DIMENSION FACT(16),ALPHAW(6),F(4)*G(4)*C(1000)
      DATA FACT
      1/1.          ,2.          ,6.          ,24.          ,
      2 120.         ,720.         ,5040.        ,4.032E4      ,
      3 3.6288E5    ,3.6288E6   ,3.99168E7  ,4.790016E8  ,
      4 6.2270208E9 ,8.71782912E10 ,1.30767437E12 ,2.09227899E13 /
      EQUIVALENCE (BETA,C(427))
      INTEGER ERROR
      1 FORMAT(18H0** FALPHA ** SUBROUTINE SPEAKING. ALPHA CANNOT BE REDU
     ICED IN 10 ITERATIONS. ALPHA = 1PE15.7)
      ERROR = 1
      I = 0
      F(1) = .166666667
      F(2) = .5
      ALPHAW(1) = ALPHA
      DUM = ABS(ALPHA)
      IF(DUM .LT. 1.E-7) GO TO 140
      IF(DUM .LT. 1.) GO TO 100
C
C ALPHA .GE. 1.
      DO 10 I = 1,10
      DUM = DUM * 0.25
      IF(DUM .LT. 1.) GO TO 20
      10 CONTINUE
      WRITE(6,1) ALPHA
      ERROR = 2
      RETURN
      20 ALPHAW(1) = SIGN(DUM,ALPHA)
C
C POWERS OF ALPHA
      100 DO 120 KMAX = 2,6
      ALPHAW(KMAX) = ALPHAW(KMAX - 1) * ALPHAW(1)
      IF(ABS(ALPHAW(KMAX)) .LT. 1.E-7) GO TO 121
      120 CONTINUE
      KMAX = 6
C
C F(I) SERIES
      121 L = 0
      123 IDUM1 = 2 - L
      IDUM2 = 2 + L
      DO 125 K = 1,KMAX
      IDUM3 = 2 * K + IDUM2
      125 F(IDUM1) = F(IDUM1) + ALPHAW(K)/FACT(IDUM3)
      L = L + 1
      GO TO (123,140), L
      140 F(3) = 1. + ALPHAW(1) * F(1)
      F(4) = 1. + ALPHAW(1) * F(2)
      IF(I .EQ. 0) GO TO 151
C
C REDUCTION FORMULI
      DO 150 K = 1,I
      F(1) = 0.25 * (F(1) + F(2) * F(3))
      F(2) = 0.5 * F(3)**2
      150 F(3) = 1. + ALPHAW(1) * F(1)
C
C G(I)
      151 DO 155 I = 1,3
      IDUM = 4 - I
      155 G(1) = BETA **I * F(IDUM)
      G(4) = F(4)
      RETURN
      END

```

```

$IBFTC ROT      DECK
C
C THIS SUBROUTINE ROTATES A VECTOR (XVEC) ABOUT ANOTHER VECTOR (AVEC)
C THROUGH AN ANGLE (EPS) AND PUTS THE RESULT IN XPR
C
SUBROUTINE ROTATE (XVEC, AVEC, EPS, XPR)
DIMENSION XVEC(4), AVEC(4), XPR(4)
A2 = AVEC(1)**2 + AVEC(2)**2 + AVEC(3)**2
SEA = SIN(EPS)/SQRT(A2)
DUM1 = (AVEC(1) * XVEC(1) + AVEC(2) * XVEC(2) + AVEC(3) * XVEC(3))
1/A2
DUM2 = -2. * SIN(EPS/2.)**2
XPR(1) = AVEC(2) * XVEC(3) - XVEC(2) * AVEC(3)
XPR(2) = XVEC(1) * AVEC(3) - AVEC(1) * XVEC(3)
XPR(3) = AVEC(1) * XVEC(2) - XVEC(1) * AVEC(2)
DO 100 I = 1,3
100 XPR(I) = DUM2 * (XVEC(I) - DUM1 * AVEC(I)) + SEA * XPR(I)
1+ XVEC(I)
XPR(4) = SQRT(XPR(1)**2 + XPR(2)**2 + XPR(3)**2)
RETURN
END

```

```

$IBFTC PLANET DECK
      SUBROUTINE PLANET(M,ERROR)
      COMMON /EPHEM/ XJD,SRV,SVV
      1          /DATA/ C
C   SUB FOR COMPUTING PLANET POS. AND VELS. BY MEAN ELEMENTS
C
C   TIME INTERVALS
C
C   2439500.5 - L=1 - 2441324.5      1/10/67 TO 1/08/72
C   2441324.5 - L=2 - 2443148.5      1/08/72 TO 1/05/77
C   2443148.5 - L=3 - 2444972.5      1/05/77 TO 1/03/82
C   2444972.5 - L=4 - 2446796.5      1/03/82 TO 1/01/87
C
C   PLANETARY BODIES
C
C   M =
C   1  SUN
C   2  MERCURY
C   3  VENUS
C   4  EARTH
C   5  MOON
C   6  MARS
C   7  JUPITER
C   8  SATURN
C   9  URANUS
C  10  NEPTUNE
C  11  PLUTO
C
C
1 FORMAT(83H0** PLANET ** SPEAKING. I DO NOT HAVE AN EPHEMERIS FOR
1MERCURY OR THE MOON. SORRY)
510 FORMAT(75H0** PLANET ** SPEAKING. TIME IS OUTSIDE THE LIMIT OF TH
1E EPHEMERIS. XJD = 1PD25.15,10H JUL. DAYS)
      INTEGER ERROR
      DOUBLE PRECISION TT(5),XJD,TEPOCH
      DIMENSION AP(3),BP(3),SVV(3),SRV(3),C(1000)
      DATA DELC2,(TT(I),I=1,5)
      1/ 912.           ,2439500.5D0   ,2441324.5D0   ,2443148.5D0   ,
      1 2444972.5D0     ,2446796.5D0   /
      IF((M .NE. 2) .OR. (M .NE. 5)) GO TO 200
      ERROR = 2
      WRITE(6,1)
      RETURN
200 ERROR = 1
C
C   FIND TIME INTERVAL
      IF(XJD.LT.TT(1) .OR. XJD.GE. TT(5)) GO TO 503
      L = SNGL(XJD - TT(1))/1824. + 1.
      TEPOCH = TT(L) + DBLE(DELC2)
C
C   DEFINITION OF PLANETARY PARAMETERS
      IAP = 456 + M * 36 + L * 9
      AP(1) = C(IAP)
      AP(2) = C(IAP + 1)
      AP(3) = C(IAP + 2)
      BP(1) = C(IAP + 3)
      BP(2) = C(IAP + 4)

```

```

      BP(3) = C(IAP + 5)
C  MEAN MOTION
      SNP = C(IAP + 6)
C  MEAN ANOMALY
      SGZP = C(IAP + 7)
C  ECCENTRICITY
      SEP = C(IAP + 8)
C
C  EVALUATE POSITION AND VELOCITY
C
300  XJP = SNGL (XJD -TEPOCH )/ DELC2
      SGP = SGZP + SNP * DELC2 * XJP
      E=SGP
      SNE= SIN(SGP)
      CSE= COS(SGP)
C
C  SOLVE KEPLER
      DO 380 I=1,2
      ESE=SNE*SEP
      ECE=1.-CSE*SEP
      BK=(SGP-E+ESE)/ECE
      E=E+BK-.5*BK*BK*ESE/ECE
      CSE= COS(E)
380  SNE= SIN(E)
C
      ADR = 1./ (1.-SEP *CSE )
      CK1=CSE-SEP
      CK2=-SNE*ADR
      CK3=CSE*ADR
      DO 390 I=1,3
      SRV(I)=AP(I) *CK1+BP(I) *SNE
390  SVV(I) = (AP(I) *CK2 + BP(I) *CK3) *SNP
      RETURN
503  ERROR = 2
      WRITE(6,510) XJD
      RETURN
      END

$DATA
1JUPITER PROBE
      Y F C   4 1 1                           EARTH TO SUN ARC
      401.12777285E4
      402.60337410E4
      403.22448028E4
      407-.1465573E2
      408.22651264E1
      409.22535885E1
      164.80299558E6
      224.48153729E8
      4256912.30
      *4260138.11                               TIME COAST ARC
      Y Y F C   1 4
      *428365.78981E0                           SUN TO JUPITER ARC
      Y N F C   1 1 7
      * Y N F C   7 5                           JUPITER REF

```

